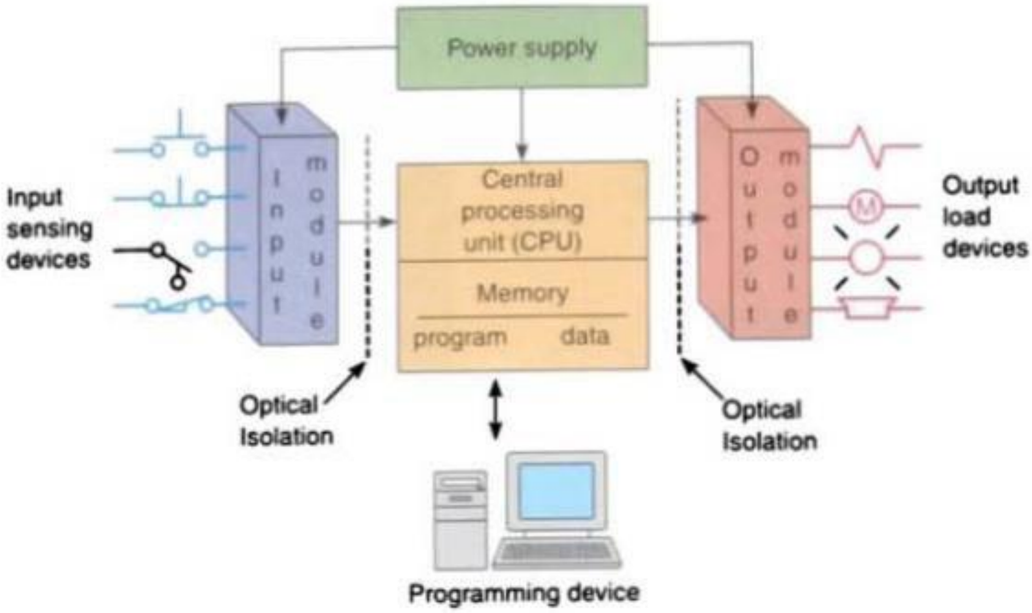
Chapter-1

Programmable Logic Controller:

Introduction:

A programmable logic controller (PLC) is a specialized Programmable device which is used to control machines and processes. It uses a programmable memory to store instructions and execute specific functions that include on/off control, timing, counting, sequencing, arithmetic, and data handling.

PLC Block diagram



Central Processing Unit

It is heart of the PLC. CPU is used to store the program, reads the status of inputs through the input module and execute the stored program and appropriate output to be activated based on the logic CPU has two memory section one section used to store the program and other section is used to store the data.

Input Module

- The I/O system forms the interface by which field devices are connected to the controller.
- The purpose of this interface is to condition the various signals received from or sent to external field devices.
- Input devices such as pushbuttons, limit switches, sensors. Selector switches and thumbwheel switches are hardwired to terminals on the input modules.

Output Module

- Output devices such as small motors, motor starters, solenoid valves and indicator lights are hardwired to the terminals on the output modules.

Memory system: This stores the program, the data, and the configuration of the PLC. The memory system can be divided into ROM (Read-Only Memory), RAM (Random Access Memory), and EEPROM (Electrically Erasable Programmable Read-Only Memory). ROM contains the operating system and the firmware of the PLC. RAM is used for temporary storage of data and variables during program execution. EEPROM is used for permanent storage of the user program and the PLC settings.

Programming device

 The programming device is used to enter the desired program into the memory of processor.

 Ladder logic programming language uses instead of words, graphic symbol...

 It is a special language written.

Power supply: This provides the necessary voltage and current to the PLC and its I/O modules. The power supply can be AC or DC, depending on the PLC model and the application requirements.

Some of the advantages of PLCs are:

i. They can replace hard-wired relay logic systems that are difficult to modify and troubleshoot.
ii. They can handle complex and sequential logic with less memory and processing power than other computers.
iii. They can interface with other devices and protocols using various I/O modules and communication standards.
iv. They can be programmed using different languages, such as Ladder Diagram, Function Block Diagram, Structured Text, Instruction List, and Sequential Function Chart.

Some of the disadvantages of PLCs are:

i. They can be expensive and require specialized software and hardware to program and operate.
ii. They can be vulnerable to electrical noise, power surges, and electromagnetic interference that can affect their performance and data integrity.
iii. They can have compatibility issues with different PLC platforms and devices, especially when using proprietary protocols and formats.
iv. They can have limited programming capabilities and features compared to general-purpose computers.

PLC applications

i. PLCs are used to automate and control various processes and machines in different industries and sectors. Some of the examples of PLC applications are
ii. Washing machines: PLCs can control the water level, temperature, cycle time, and spin speed of washing machines, as well as detect faults and errors1.
iii. Traffic light systems: PLCs can coordinate the timing and sequence of traffic lights, as well as adjust them according to traffic conditions and emergency situations.
iv. Car washes: PLCs can control the water jets, brushes, dryers, and conveyor belts of car washes, as well as monitor the sensors and safety devices.
v. Luggage handling systems: PLCs can control the sorting, routing, and tracking of luggage in airports, as well as communicate with barcode scanners and display boards.

vi. Automatic doors: PLCs can control the opening and closing of doors based on the presence and direction of people, as well as the ambient light and temperature.

These are just some of the examples of PLC applications. PLCs can also be used for many other purposes, such as industrial batch washing machines, water tank quenching systems, fire detection systems, smart traffic control systems, and more. PLCs are versatile and reliable devices that can perform various functions and operations in different environments.

PLC programming languages are the methods of creating programs for programmable logic controllers (PLCs), which are devices that control industrial processes and machines. PLC programming languages are specifies five types of languages: Ladder Diagram (LD), Function Block Diagram (FBD), Structured Text (ST), Instruction List (IL), and Sequential Function Chart (SFC)1.

Each PLC programming language has its own advantages, disadvantages, and best uses cases. Some of the factors that influence the choice of PLC programming language are the complexity of the application, the readability and maintainability of the code, the compatibility with the PLC platform, and the preference and experience of the programmer2.

The five PLC programming languages are:

Ladder Diagram (LD): This is the most popular and widely used PLC programming language. It is based on the graphical representation of relay logic circuits, which use contacts, coils, and special instructions to implement logic. LD is easy to understand and troubleshoot, especially for electricians and technicians who are familiar with relay wiring diagrams. However, LD can be inefficient and cumbersome for complex applications, as it requires a lot of memory and processing power.

Function Block Diagram (FBD): This is a graphical PLC programming language that uses blocks to represent functions, operations, and variables. The blocks are connected by wires that carry data between them. FBD is suitable for implementing control systems that are based on Piping and Instrumentation Diagrams (P&ID), such as chemical processes. FBD is also useful for creating reusable and modular code. However, FBD can be difficult to read and debug, especially when the diagrams become large and complex.

Structured Text (ST): This is a textual PLC programming language that uses high-level syntax similar to Pascal, C, or Java. ST is powerful and flexible, as it allows the programmer to use variables, data types, expressions, statements, and control structures. ST is ideal for implementing complex algorithms and calculations, as well as interfacing with other devices and protocols. However, ST can be challenging to learn and master, especially for beginners and those who are used to graphical languages. ST can also be less intuitive and transparent than LD or FBD.

Instruction List (IL): This is a textual PLC programming language that uses low-level syntax similar to assembly language. IL is composed of instructions that operate on operands, which can be constants, variables, or memory addresses. IL is efficient and compact, as it allows the programmer to directly manipulate the bits and bytes of the PLC memory. IL is also compatible with most PLC platforms and devices. However, IL can be hard to read and maintain, as it requires the programmer to memorize the instruction mnemonics and the operand addresses. IL can also be prone to errors and bugs, as it does not have any error checking or debugging features.

Sequential Function Chart (SFC): This is a graphical PLC programming language that uses steps, transitions, and actions to represent the sequential behavior of a system. The steps are the states of the system, the transitions are the conditions for changing the states, and the actions are the activities that occur in each state. SFC is useful for modeling and documenting the logic and flow of a system, as well as for dividing the system into manageable parts. However, SFC can be limited and restrictive, as it does not allow the programmer to use variables, data types, or expressions. SFC can also be incompatible with some PLC platforms and devices.

Advantages of PLC over relay

- Doesn't have complex wiring
- It can act as a timer, comparator, and counter
- It can handle the complex process
- Hardware failure is very less when compared to a relay logic
- In order to change the logic in relay there should be panel rewiring but in the case of PLC this is not necessary
- Industrial process can be carried out securely by using PLC
- PLCs require less hardware when compared to the relay and due to this, the complexity would be reduced.
- The faults related to the complex hardware would be reduced due to the less wiring
- PLC is really compact and due to this it would only take less space
- PLCs have memory and better capabilities than the relay

Major difference between PLC and relay

PLC	RELAY
Very less wiring when compared to relay logic	It requires a lot of wiring
Logic can be easily changed by changing the program	Rewiring is required to change the logic
Easily replaced	Relay won't handle the environmental conditions as well as the PLC

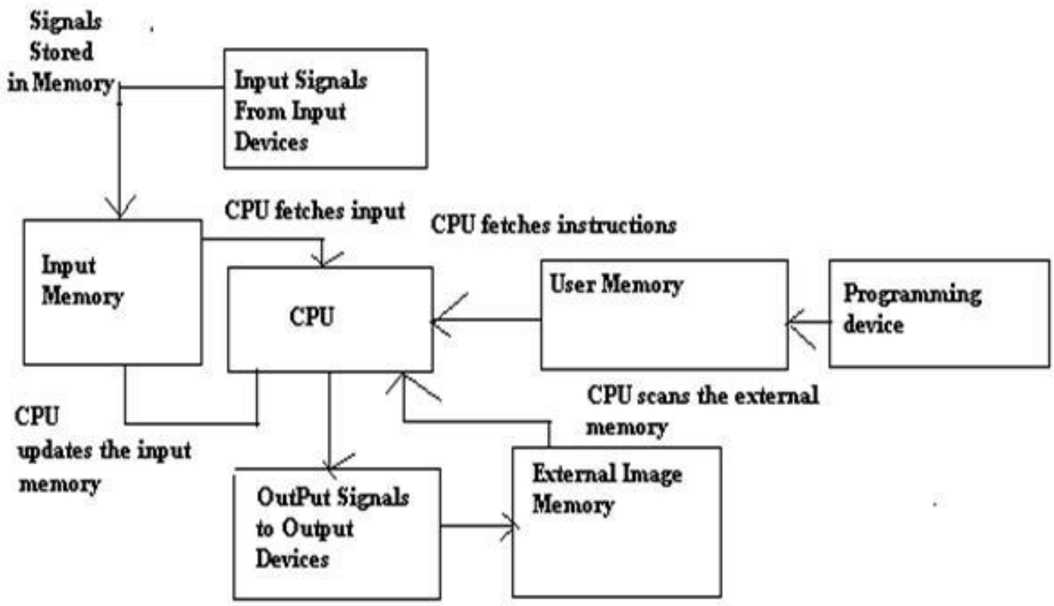
It can handle environmental conditions	Relay won’t handle the environmental conditions as good as the PLC
Compact size	It’s not compact and due to this it would take a lot of space
PLC’s have memory	Relays don’t have memory
Good reliability	Less reliable due to the moving parts
Faults can be easily cleared	It would take a lot of time to find out the faults so faults can’t be easily cleared
Computational capability is high	Very less computational capability
PLCs can handle vibrations, temperature, humidity etc	Relays can’t handle humidity, vibrations, and temperature as good as PLC’s

Leading manufacturer for PLC

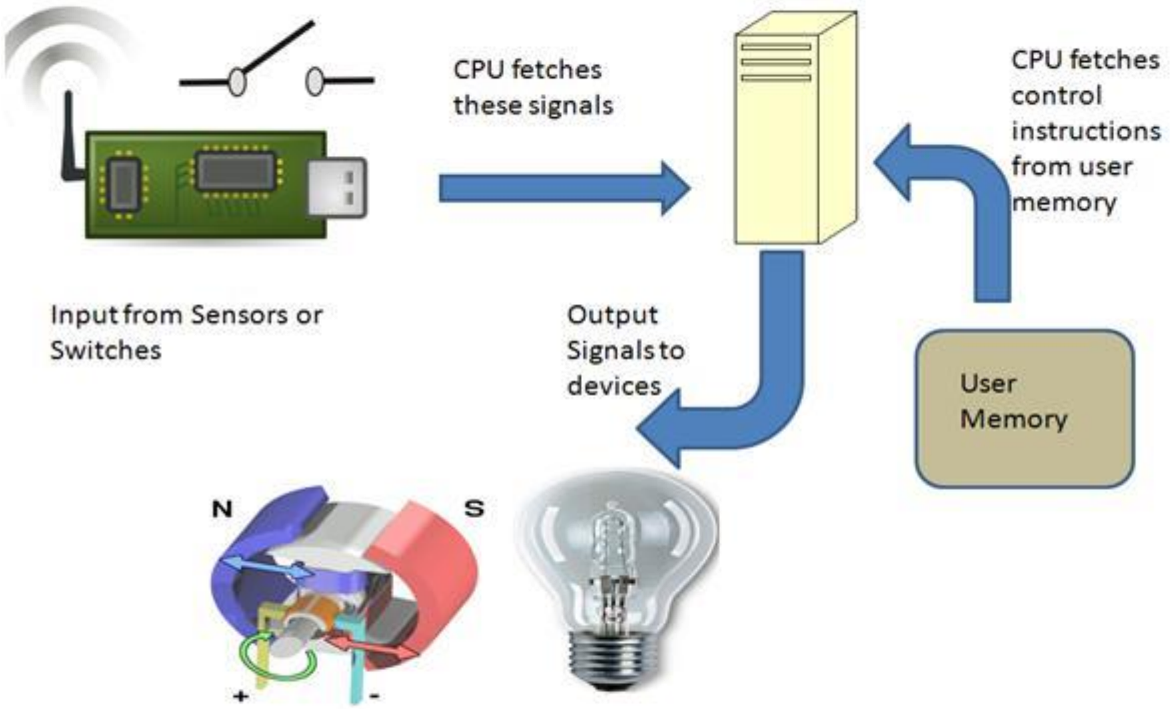
- Allen Bradley
- ABB
- Siemens
- Mitsubishi PLC
- Hitachi PLC
- Delta PLC
- General Electric (GE) PLC
- Honeywell PLC

Chapter-2

Working of a PLC



Working PLC Schematic



Working of PLC

- The input sources convert the real-time analog electric signals to suitable digital electric signals and these signals are applied to the PLC through the connector rails.
- These input signals are stored in the PLC external image memory in locations known as bits. This is done by the CPU
- The control logic or the program instructions are written onto the programming device through symbols or through mnemonics and stored in the user memory.

- The CPU fetches these instructions from the user memory and executes the input signals by manipulating, computing, processing them to control the output devices.
- The execution results are then stored in the external image memory which controls the output drives.
- The CPU also keeps a check on the output signals and keeps updating the contents of the input image memory according to the changes in the output memory.
- The CPU also performs internal programming functions like setting and resetting of the timer, checking the user memory.

PLC working principle

Let's look into how PLC is working. PLC's CPU continuously scans its program which is programmed by basically ladder logic. Unlike other programming languages, ladder programming scanning sequence is a little bit different. First, CPU scans the inputs and then it executes the program. After executing the program, the CPU updates the outputs. This scanning sequence is illustrated in Figure.

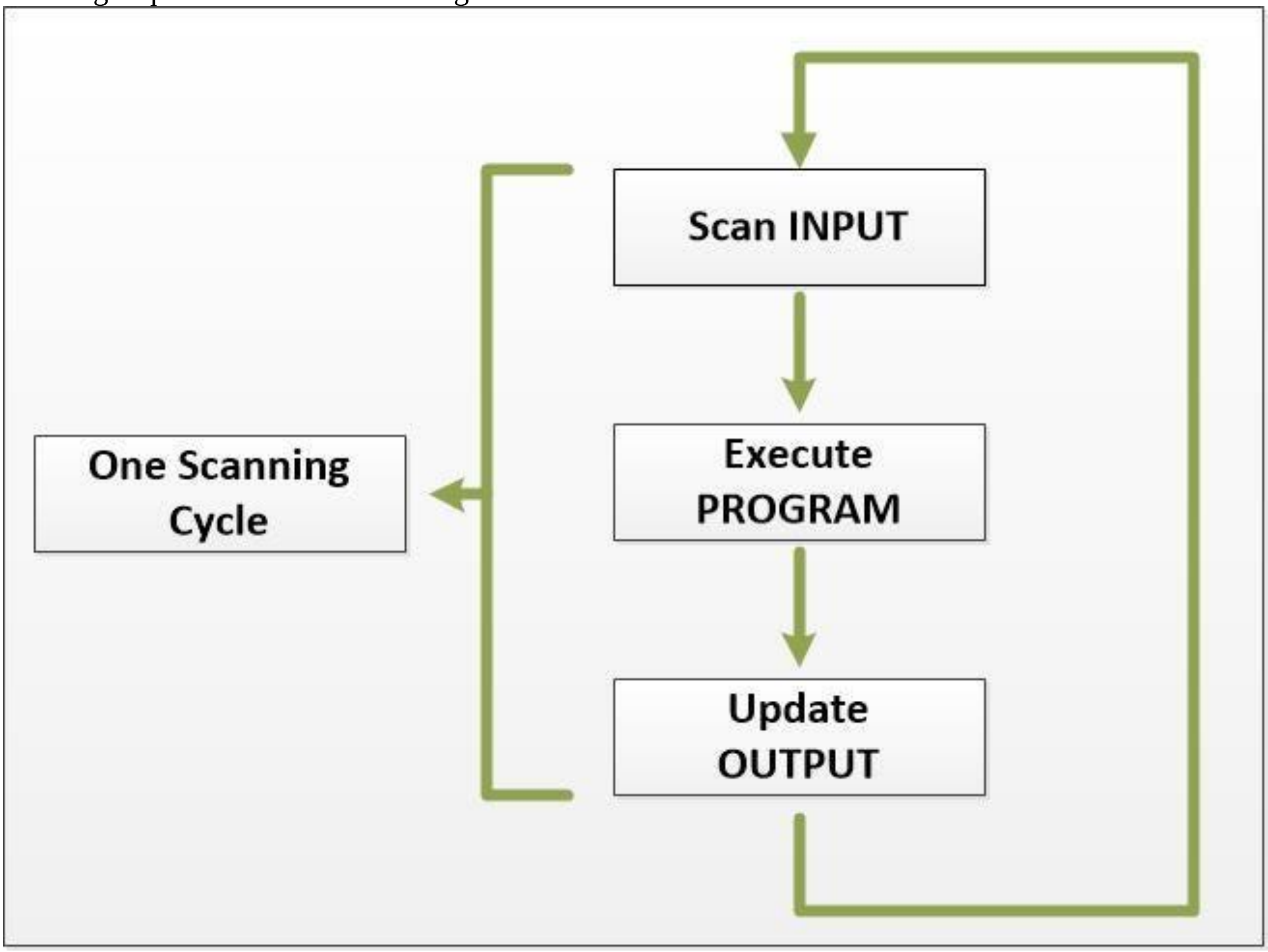


Fig: PLC Scanning Sequence

When it comes to ladder logic scanning, PLC scans its program top to bottom and rungs in the program are scanned from left to right order. This scanning pattern is shown in figure 1.5.

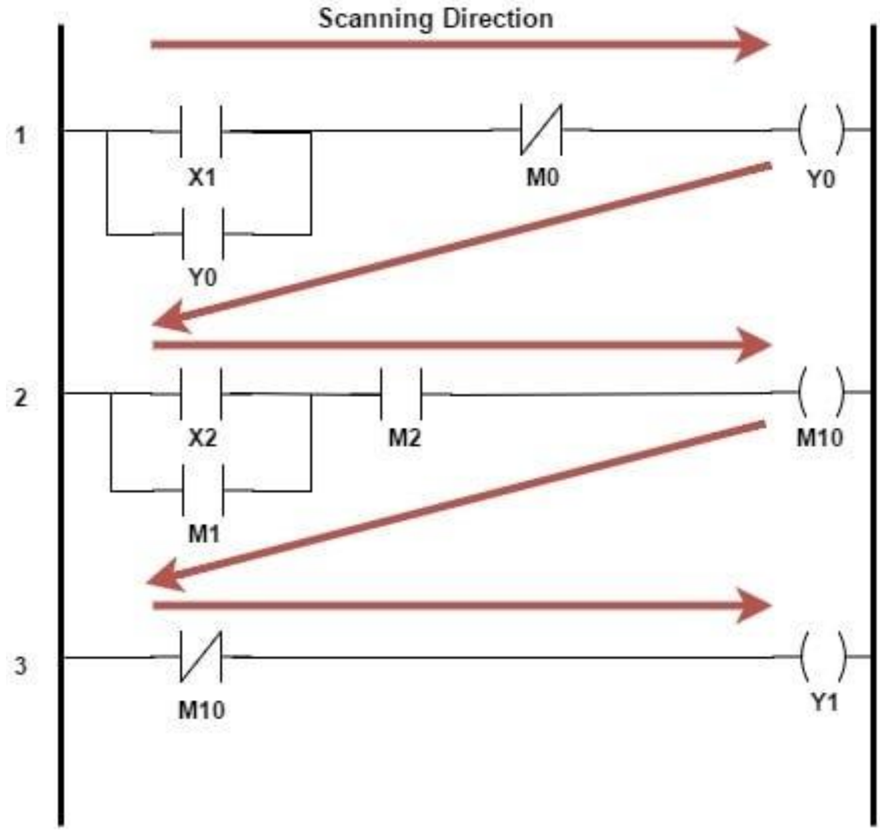


Fig: Ladder logic scanning sequence

The Basics of PLC Operation

The operation of a PLC is very simple. The processor makes decisions based on a ladder logic program written by the user. In order to use the program properly, the PLC must communicate with the various field devices it is tasked with monitoring and controlling. It then compares the actual conditions of the field devices with what the program instructs them to do, and updates the output devices accordingly.

Operational Sequence

The operational sequence is as follows:

1. Input switch is pressed.
2. Input module places a “1” in the input data table,
3. The ladder logic program sees the “1” and caused a “1” to be put into the output data table.
4. The output data table causes the output module to energize associated point.
5. The output device energizes.

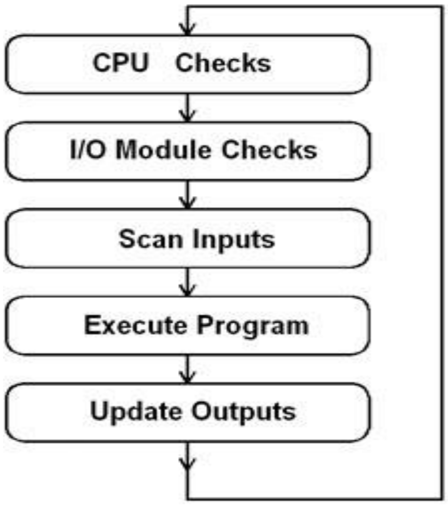
The Scan Cycle

PLCs operate by continually scanning programs and repeat this process many times per second. When a PLC starts, it runs checks on the hardware and software for faults, also called a self-test. If there are no problems, then the PLC will start the scan cycle. The scan cycle consists of three steps: input scan, executing program(s), and output scan.

Input Scan: A simple way of looking at this is the PLC takes a snapshot of the inputs and solves the logic. The PLC looks at each input card to determine if it is ON or OFF and saves this information in a data table for use in the next step. This makes the process faster and avoids cases where an input changes from the start to the end of the program.

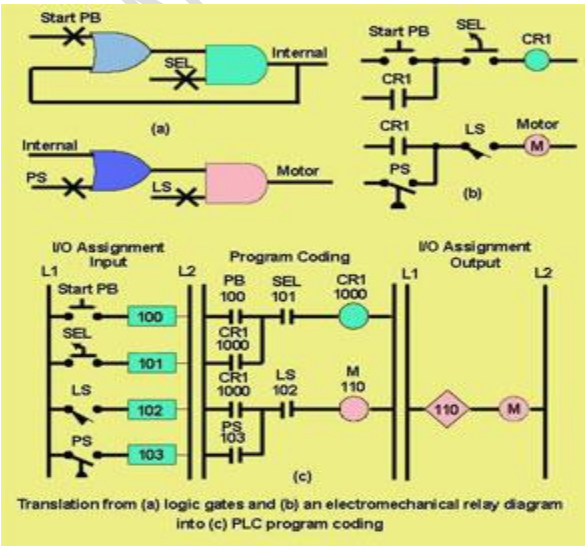
Execute Program (or Logic Execution): The PLC executes a program one instruction at a time using only the memory copy of the inputs the ladder logic program. For example, the program has the first input as ON. Since the PLC knows which inputs are ON/OFF from the previous step, it will be able to decide whether the first output should be turned ON.

Output Scan: When the ladder scan completes, the outputs are updated using the temporary values in memory. The PLC updates the status of the outputs based on which inputs were ON during the first step and the results of executing a program during the second step. The PLC now restarts the process by starting a self-check for faults.



PLC Scan Cycle

Logic Scan: Ladder logic programs are modeled after relay logic. In relay logic, each element in the ladder will switch as quickly as possible. Program elements can only be examined one at a time in a fixed sequence. The ladder logic scan begins at the top rung. At the end of the rung, it interprets the top output first, and then the output branched below it. On the second rung, it solves branches, before moving along the ladder logic rung.



Translation from (a) logic gates and (b) an electromechanical relay diagram into (c) PLC program coding

Chapter 3

What is a Ladder Diagram?

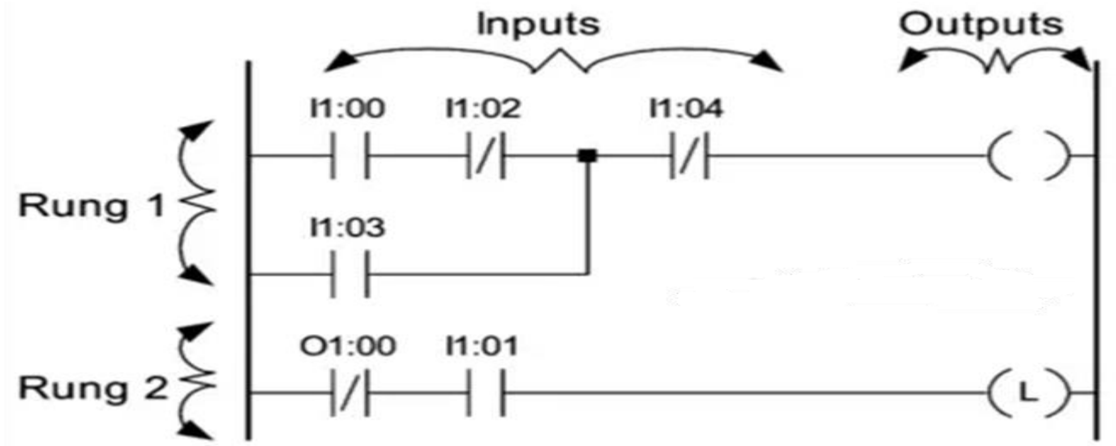
A Ladder Diagram is one of the simplest methods used to **program a PLC**.

It is a graphical programming language that evolved from electrical relay circuits.

Each program statement is represented with a line, called the rung, that has all relevant inputs to the left and the output to the right.

The **output device** of a rung is energized if electric power can conceptually flow from the left side of the rung to the right side. Input devices are assumed to block the flow of power if they are not activated.

During the execution of a **ladder diagram**, the PLC reads the states of all inputs, then determines the states of all outputs starting from the rung at the top side, going down to the last rung, and finally updates the state of the output devices.



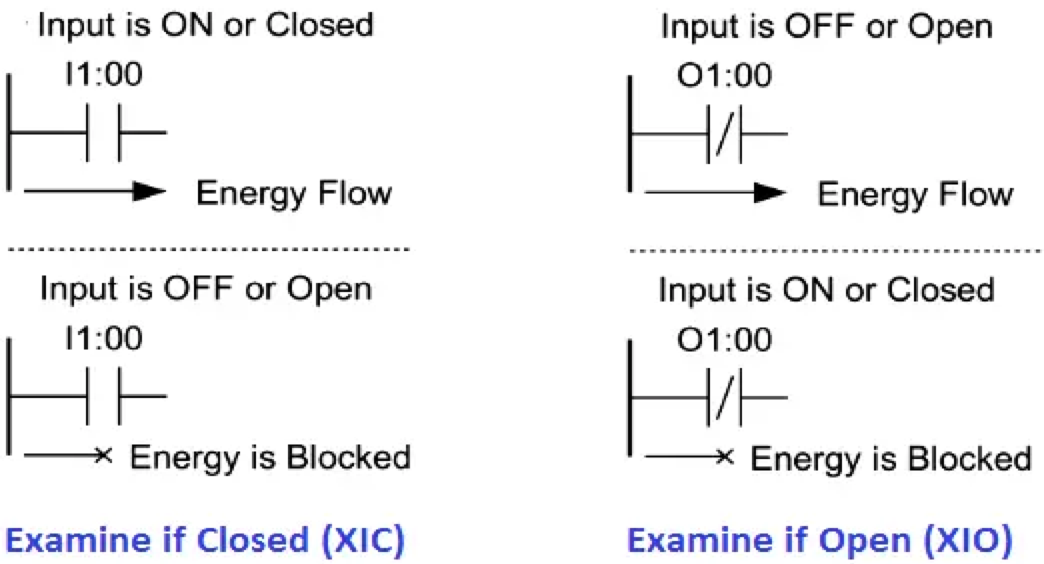
Naming Convention

During the development of a PLC program, we must use specific names to identify the inputs, outputs, memory flags, timers, and counters.

PLC manufacturers use a variety of approaches in naming the inputs, outputs and other resources.

A typical naming convention is to identify inputs with the letter "I" and outputs with the letter "O", followed by a 1-digit number that identifies the slot number and a 2-digit number that identifies the position of the input or output in the slot.

Relay Logic Instructions



Examine if Closed (XIC)

If the input device is ON or Closed, then the corresponding bit in the data memory (input image) is set to true, thus allowing (conceptually) the energy to flow from its left side to its right-hand side.

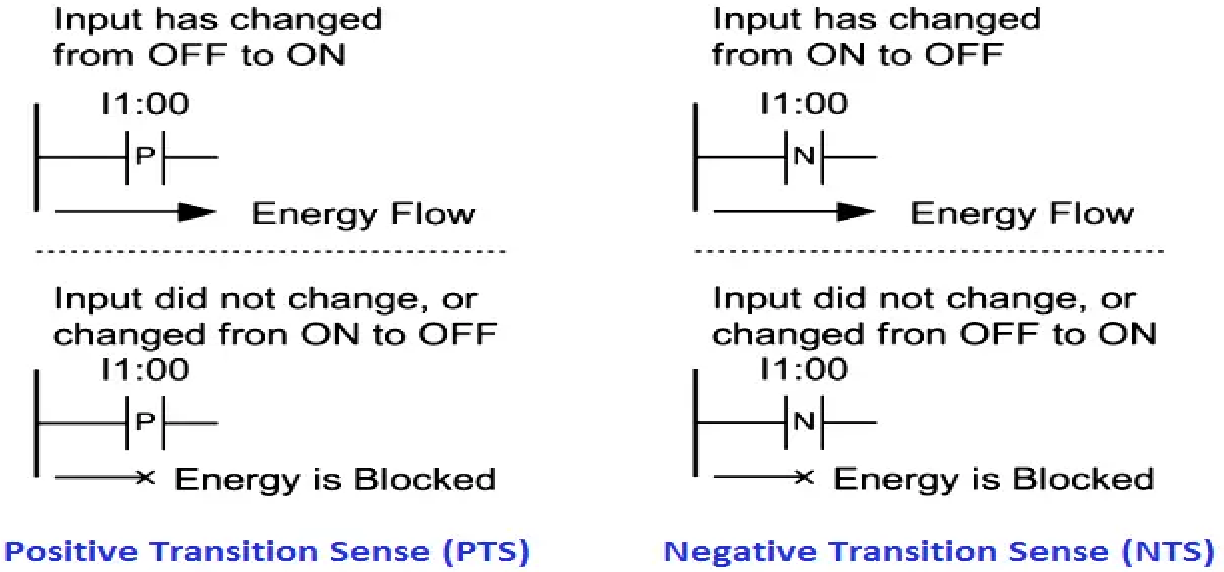
Otherwise, it is set to false, thus blocking the energy.

Examine if Open (XIO)

If the input device is OFF or Open, then the corresponding bit in the data memory (input image) is set to true, thus allowing (conceptually) the energy to flow from its left side to its right-hand side.

Otherwise, it is set to false, thus blocking the energy.

Input Transition Sensing Instructions



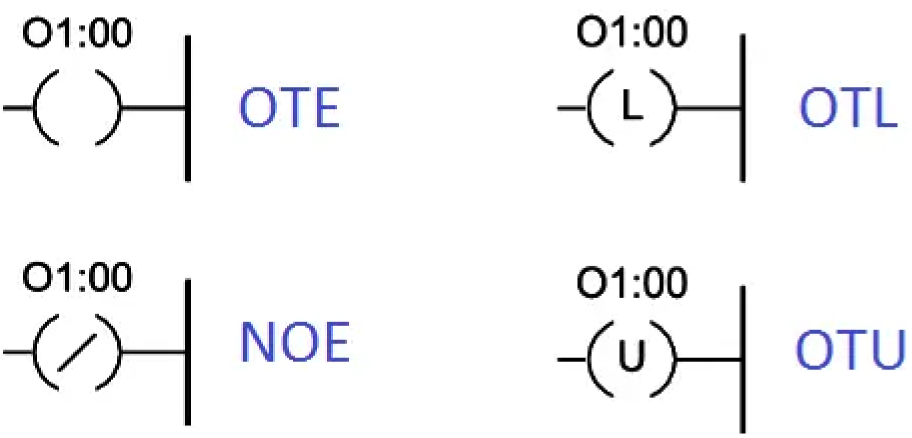
Positive Transition Sense (PTS)

The condition of the right link is ON for one ladder rung evaluation when a change from OFF to ON at the specified input is sensed.

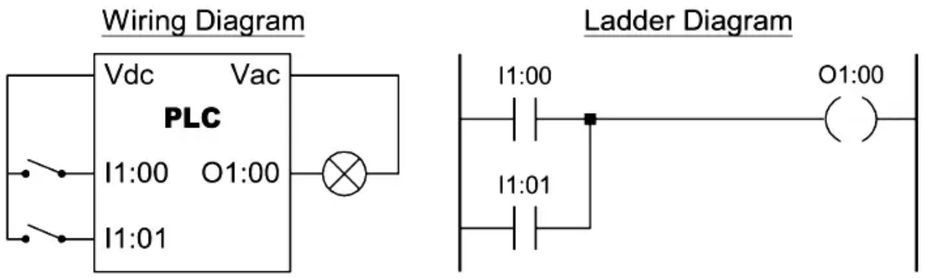
Negative Transition Sense (NTS)

The condition of the right link is ON for one ladder rung evaluation when a change from ON to OFF at the specified input is sensed.

Output Instructions



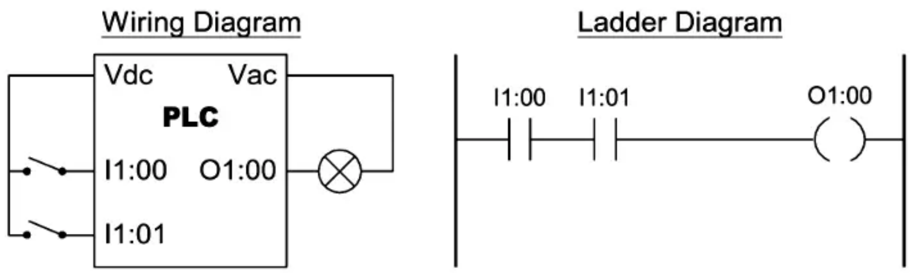
Output Energize (OTE)
If the condition of the left link of the OTE is ON then the corresponding bit in the output data memory is set. The device wired to this output is also energized.
Negative Output Energize (NOE)
If the condition of the left link of the OTE is OFF then the corresponding bit in the output data memory is set. The device wired to this output is also energized.
Output Latch/Set and Output Unlatch/Reset (OTL), (OTU)
If the condition of the left link of the OTL is momentary ON then the corresponding bit in the output data memory is set, and remains set even if the condition switches to the OFF state. The output will remain set until the condition of the left link of the OTU is momentary ON.
Basic Logic Functions
Two Input OR Function



Truth Table

I1:00	I1:01	O1:00
OFF	OFF	OFF
OFF	ON	ON
ON	OFF	ON
ON	ON	ON

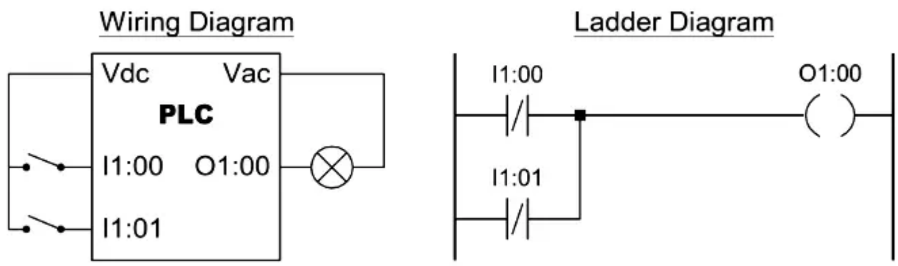
The output is ON if any of the two inputs is ON.
Two Input AND Function



Truth Table

I1:00	I1:01	O1:00
OFF	OFF	OFF
OFF	ON	OFF
ON	OFF	OFF
ON	ON	ON

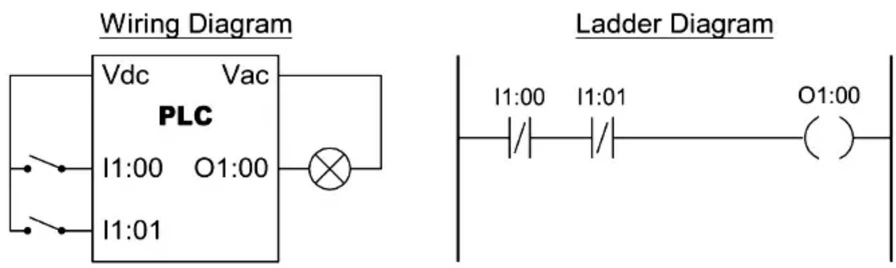
The output is ON if both of the two inputs are ON.
Two Input NAND Function



Truth Table

I1:00	I1:01	O1:00
OFF	OFF	ON
OFF	ON	ON
ON	OFF	ON
ON	ON	OFF

The output is ON if any of the two inputs is OFF.
Two Input NOR Function

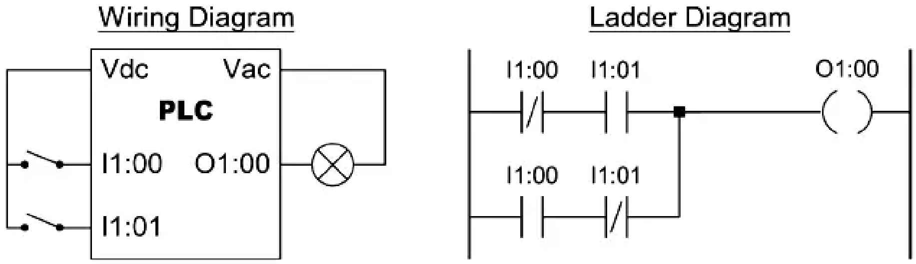


Truth Table

I1:00	I1:01	O1:00
OFF	OFF	ON
OFF	ON	OFF
ON	OFF	OFF
ON	ON	OFF

The output is ON if both of the two inputs are OFF.

Two Input EXOR Function

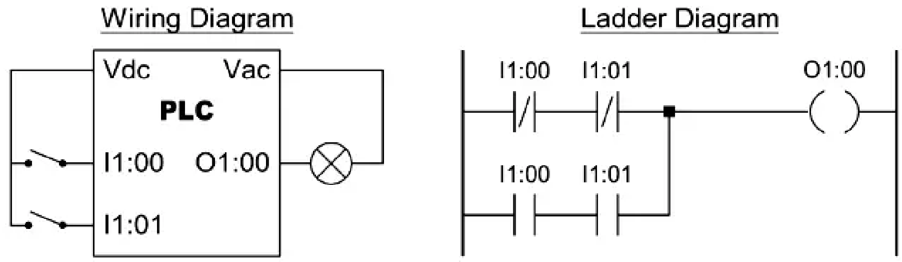


Truth Table

I1:00	I1:01	O1:00
OFF	OFF	OFF
OFF	ON	ON
ON	OFF	ON
ON	ON	OFF

The output is ON if any of the two inputs is ON, but not both.

Two Input EXNOR Function



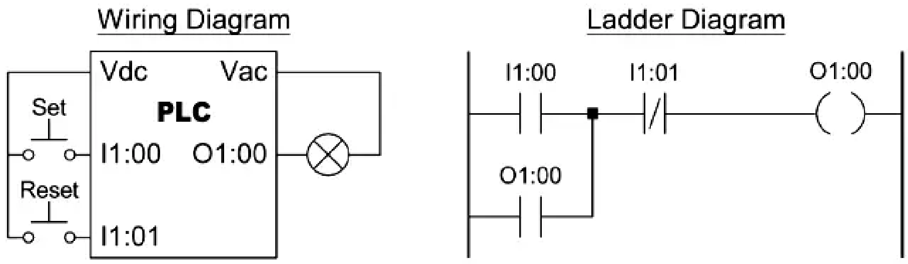
Truth Table

I1:00	I1:01	O1:00
OFF	OFF	ON
OFF	ON	OFF
ON	OFF	OFF
ON	ON	ON

The output is ON if both of the two inputs are either OFF or ON.

Set/Reset Latch Instructions

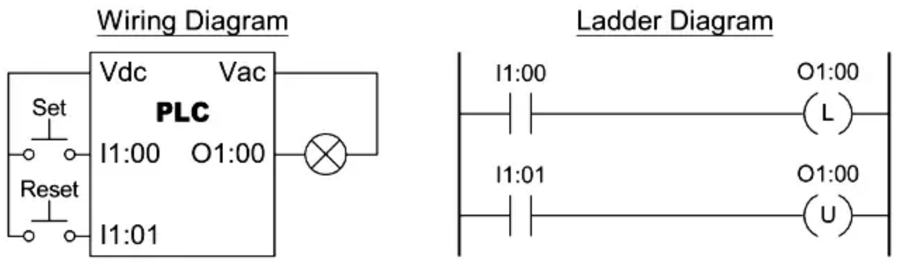
Set/Reset Latch using a Hold-in contact



Truth Table

I1:00	I1:01	O1:00
OFF	OFF	O1:00'
OFF	ON	OFF
ON	OFF	ON
ON	ON	OFF-?

Set/Reset Latch using Latch/Unlatch outputs



Truth Table

I1:00	I1:01	O1:00
OFF	OFF	O1:00'
OFF	ON	OFF
ON	OFF	ON
ON	ON	OFF-?

Notes:

- O1:00' means that the output is unchanged
- If both inputs are ON then normally the output is OFF, since the Unlatch rung appears last in the ladder diagram.

Timer Instructions

Timer Instructions are output instructions used to time intervals for which their rung conditions are true (TON), or false (TOF).

These are software timers. Their resolution and accuracy depend on a tick timer maintained by the microprocessor.

Each timer instruction has two values (integers) associated with it:

- **Accumulated Value (ACC):** This is the current number of ticks (time-base intervals) that have been counted from the moment that the timer has been energized.
- **Preset Value (PR):** This is a predetermined value set by the programmer. When the accumulated value is equal to, or greater than the preset value, a status bit is set. This bit can be used to control an output device.

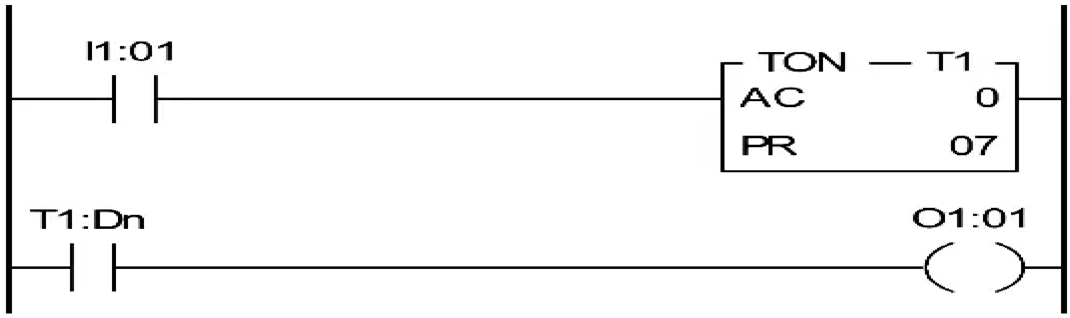
Each timer is associated with two status bits:

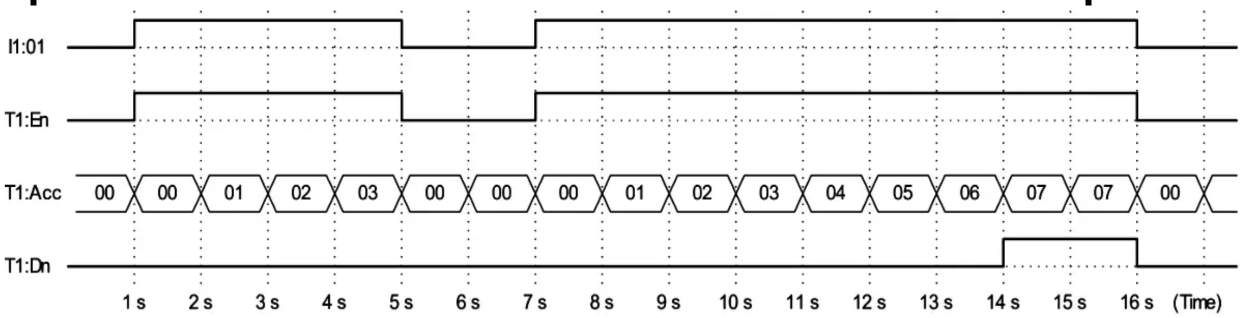
- **Timer Enable Bit (EN):** This bit is set when the rung condition to the left of the timer instruction is true. When this bit is set, the accumulated value is incremented on each time-base interval, until it reaches the preset value.
- **Done Bit (DN):** This bit is set when the accumulated value is equal to the preset value. It is reset when the rung condition becomes false.

Timer On-Delay (TON) Instruction

The TON instruction begins to count when its input rung conditions are true. The accumulated value is reset when the input rung conditions become false.

Timer ladder diagram example:

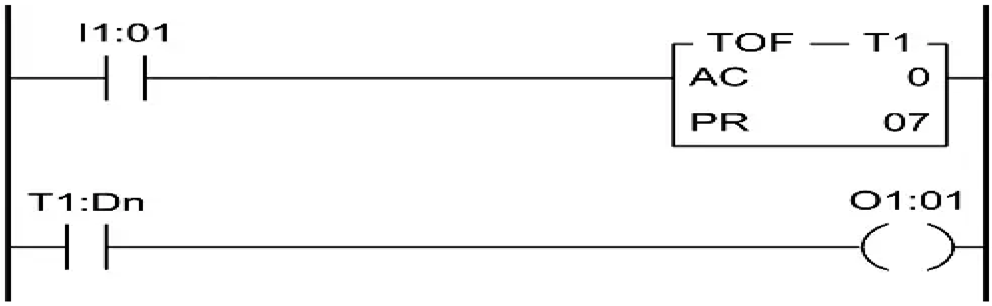




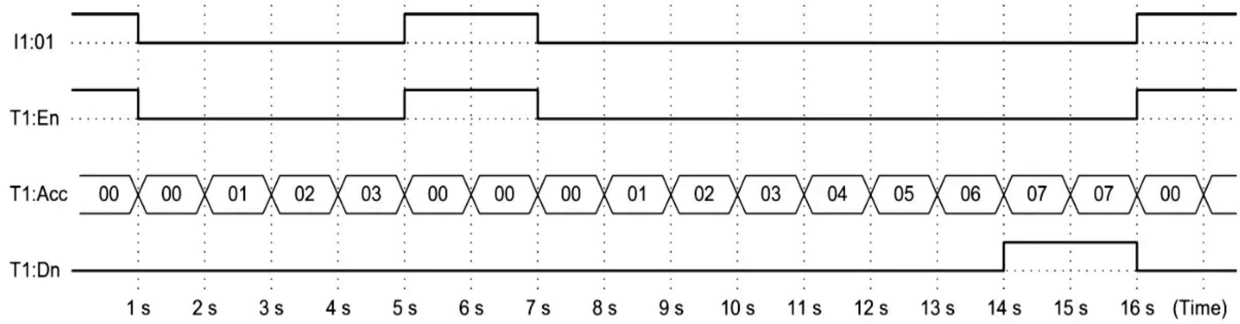
Timer Off-Delay (TOF) Instruction

The TOF instruction begins to count when its input rung makes a true-to-false transition, and continues counting for as long as the input rung remains false. The accumulated value is reset when the input rung conditions become false.

Timer ladder diagram example:



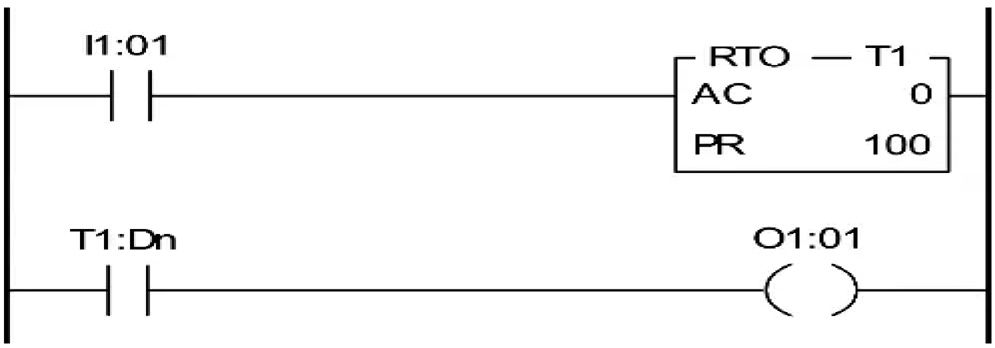
Typical timing diagram (Assume that Preset = 07)



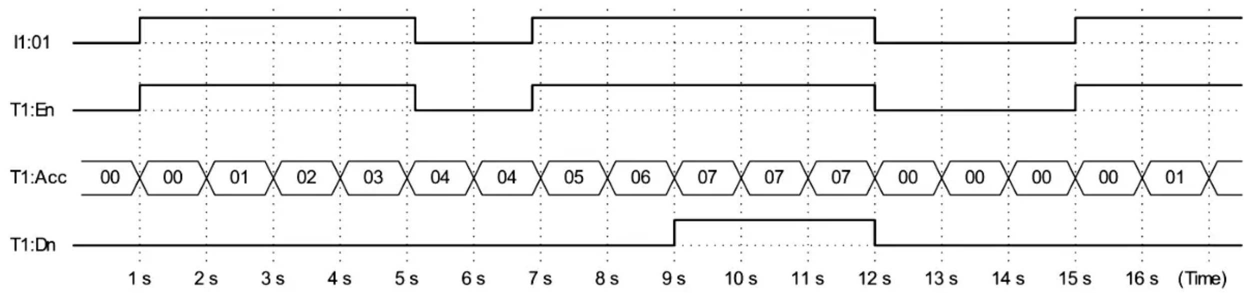
Retentive Timer (RTO) Instruction

The RTO instruction begins to count when its input rung conditions are true. The accumulated value is retained when the input rung conditions become false, and continues counting after the input rung conditions become true.

Timer ladder diagram example:



Typical timing diagram (Assume that Preset = 07)



Counter Instructions

Counter Instructions are output instructions used to count false-to-true rung transitions. These transitions are usually caused by events occurring at an input.

These counters can be UP (incrementing) or DOWN (decrementing).

Each counter instruction has two values (integers) associated with it:

- **Accumulated Value (ACC):** This is the current number of the counter. The initial value is zero.
- **Preset Value (PR):** This is a predetermined value set by the programmer. When the accumulated value is equal to, or greater than the preset value, a status bit is set. This bit can be used to control an output device.

Each counter is associated with two status bits:

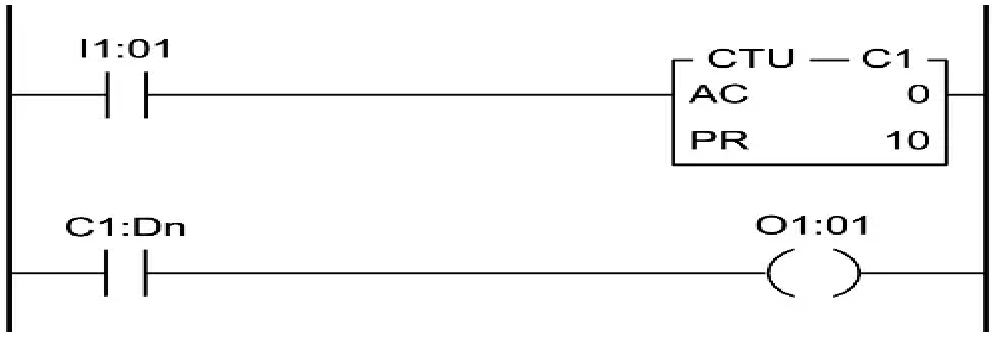
- **Counter Enable Bit (EN):** This bit is set when a false-to-true rung condition to the left of the counter instruction is detected.
- **Done Bit (DN):** This bit is set when the accumulated value is equal to the preset value. It is reset when the rung condition becomes false.

The maximum count value is 9999*. After a maximum count is reached, the counters reset and start counting from zero.

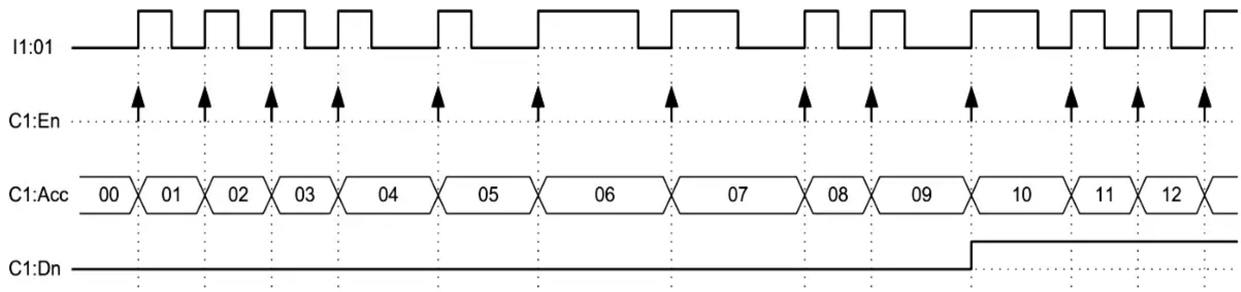
Count-up (CTU) Instruction

The CTU instruction increments its accumulated value on each false-to-true transition at its input, starting from 0.

Counter ladder diagram example:



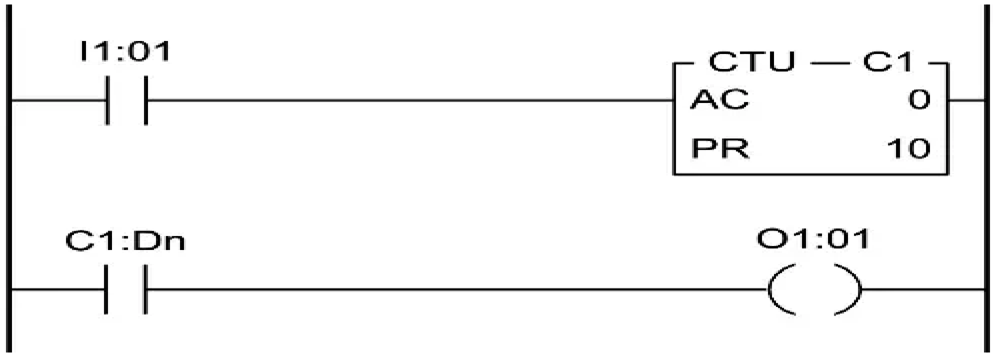
Typical timing diagram (Assume that Preset = 10)



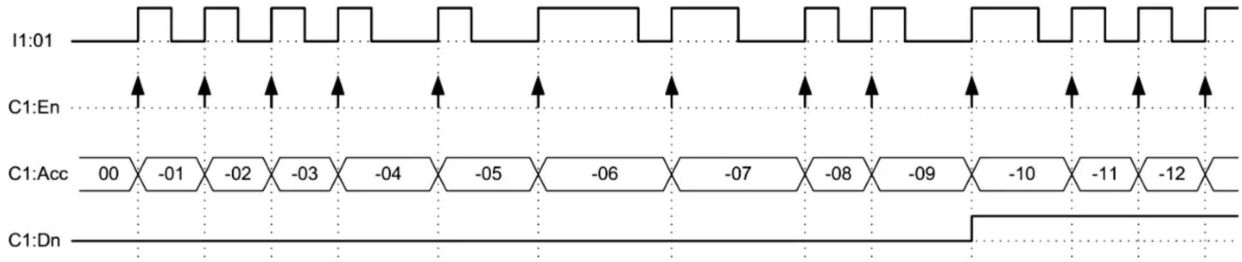
Count-down (CTD) Instruction

The CTD instruction decrements its accumulated value on each false-to-true transition at its input, starting from 0.

Counter ladder diagram example:



Typical timing diagram (Assume that Preset = -10)



The Reset (RES) Instruction

The RES instruction resets timing and counting instructions.

When the RES instruction is enabled it resets the following:

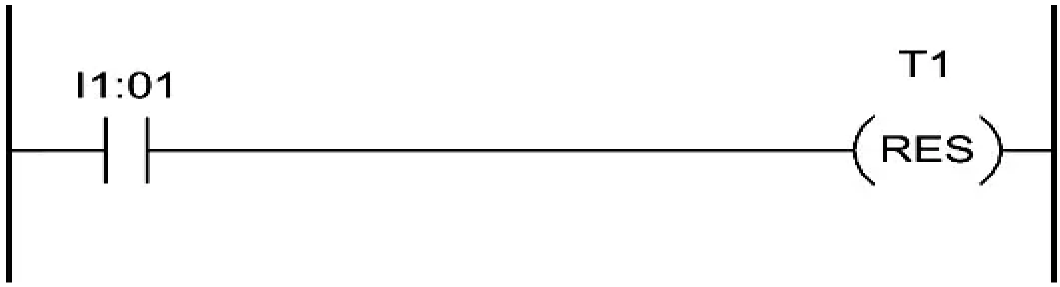
Counters:

- Accumulated value
- Counter Done Bit
- Counter Enabled Bit

Timers:

- Accumulated value
- Timer Done Bit
- Timer Timing Bit
- Timer Enable Bit

Reset ladder diagram example:



Master Control Reset (MCR) instruction:

A Master Control Reset (MCR) **instruction is an output instruction. MCR instructions are always used in pair.**

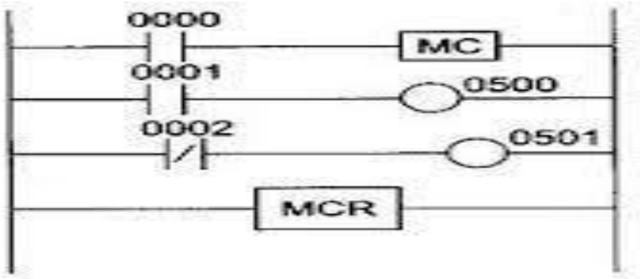
The paired instructions cause the PLC to enable or inhibit a zone of ladder logic program outputs according the application logic. The zone being controlled is known as the MCR zone control. It begins with the rung that has the first MCR instruction. The MCR zone ends with the rung that has the second MCR instruction only.

Using this instruction, we can disable all the outputs at the same when **MCR condition** goes off.

Two **MCR instruction** are used to perform the function, One is to start the MCR zone with input condition as must and another one is to end the limit area.

An input condition like Emergency switch is programmed on the rung of the first MCR to control rung logic continuity. When the rung goes “false” all non-retentive outputs within the

controlled zone are disabled. When the rung goes “true” all rungs are scanned according to their normal rung conditions.
No need to give any condition for closing **MCR zone**.



Ladder Diagram using Master Control (MC/MCR)

Some other PLC Instructions are:

- Relay-type (Basic) instructions: I, O, OSR, SET, RES, T, C
- Data Handling Instructions:
- Data move Instructions: MOV, COP, FLL, TOD, FRD, DEG, RAD (degrees to radian).
- Comparison instructions: EQU (equal), NEQ (not equal), GEQ (greater than or equal), GRT (greater than).
- Mathematical instructions.
- Continuous Control Instructions (PID instructions).
- Program flow control instructions: MCR (master control reset), JMP, LBL, JSR, SBR, RET, SUS, REF
- Specific instructions:
- BSL, BSR (bit shift justify/right), SQO (sequencer output), SQC (sequencer compare), SQL (sequencer load).
- High-speed counter instructions: HSC, HSL, RES, HSE
- Communication instructions: MSQ, SVC
- ASCII instructions: ABL, ACB, ACI, ACL, CAN

A **Distributed Control System (DCS)** is a computerized control system used in various industries for process automation.

Definition:

A DCS consists of **autonomous controllers** distributed throughout a process or plant. Unlike systems with centralized controllers, a DCS does not rely on a central operator supervisory control.

These autonomous controllers are strategically placed near the process plant, enhancing reliability and reducing installation costs. Remote monitoring and supervision complement the localized control functions.

Key Attributes:

Reliability: DCS achieves reliability by distributing control processing across nodes in the system. If a processor fails, it impacts only a specific section of the plant process, unlike a central computer failure that affects the entire process.

Fast Processing: Localizing computing power near field Input/Output (I/O) connection racks ensures rapid controller processing by minimizing network and central processing delays.

Applications:

DCS finds application in various continuous or batch-oriented processes, including:

Chemical plants

Petrochemical (oil) refineries

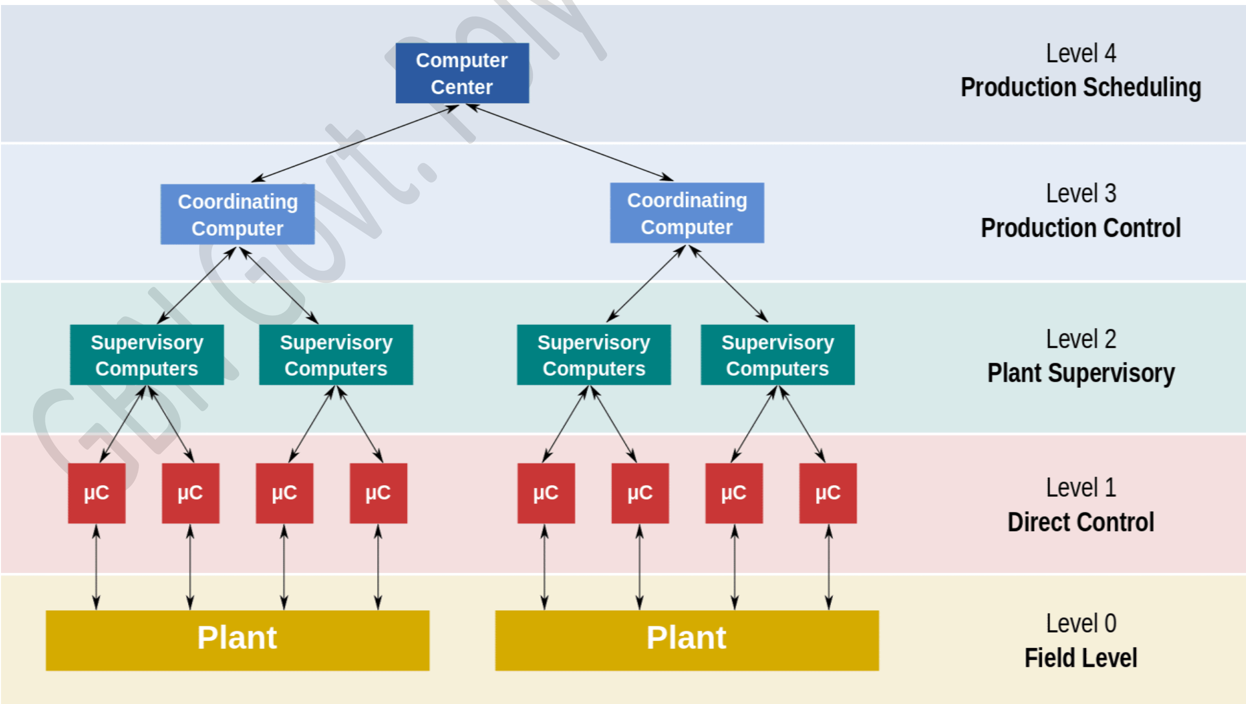
Pulp and paper mills (quality control systems)

Boiler controls and power plant systems

Nuclear power plants.

DCS architecture or block diagram:-

The devices in this process can be divided into the following five functional levels:



Level zero consists of field devices, like sensors and other control elements.

Level one is the I/O modules and processors.

Level two is the supervisory computers that collect data from processor nodes.

Level three focuses on the production control level that monitors production.

Level four includes production scheduling.

What are the advantages and challenges of a DCS?

Distributed control systems provide the following benefits:

Complex structures: Unlike comparable programmable logic controllers (PLCs), a DCS can access large amounts of information in a complex environment.

System redundancy: If a processor fails, the redundancy provided in the DCS ensures that only one section of the plant's processes is interrupted.

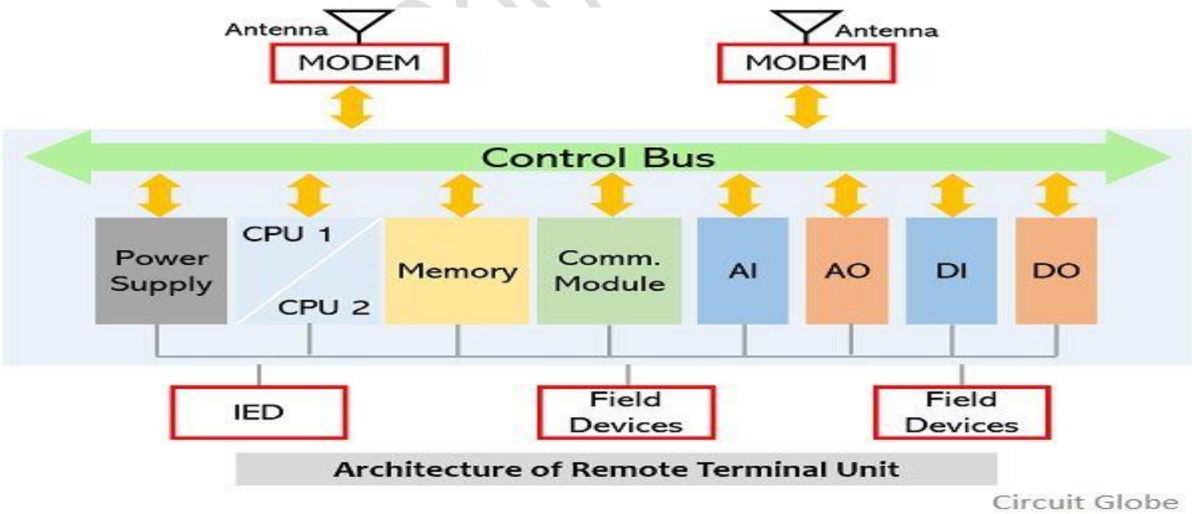
Scalability: More control or process units can be added whenever needed. Adding more I/O modules to a controller also extends I/Os.

Security: Security and cyber security capabilities are enabled at the engineer and operator levels.

Some downsides of the technology include situations where failure of one controller affects more than one loop. DCSs can also increase software development costs, and diagnosing problems can be a complex process.

REMOTE TERMINAL UNITS

A Remote Terminal Unit (RTU) is an industrial control system that is microprocessor-based. RTUs electronically connect various kinds of hardware to other control systems such as supervisory control and data acquisition (SCADA) systems or distributed control systems (DCS). These electronic units carry sensor data through input and output streams, which are transmitted to centralized industrial control systems (ICS) through a control loop. RTUs also manage connections to remote or local controls, which is why they are known as **remote units of Telecontrol or remote units of telemetry**. RTUs collect data from sensors, valves, actuators and other field devices and send that information to a master system. It also receives commands from the master system and executes them on the field devices. RTUs are used in industrial applications such as power generation, oil and gas, water treatment and transportation.



An RTU consists of a central processing unit (CPU), a power supply, a communication port and input/output (I/O) modules. The CPU is a microprocessor that performs data processing and logic functions. The power supply provides the necessary voltage and current for the RTU and its connected devices. The communication port enables data exchange with the master system using various protocols such as RS-232, RS-485 or Ethernet. The I/O modules interface with the field devices using analog or digital signals.

DCS vs. programmable logic controller systems

PLCs are another type of industrial control technology. They are small, modular computers with customized instructions for performing a particular task. Like DCSs, PLCs are used in ICSs for a variety of industries. PLCs were also designed to replace mechanical relays, drum sequencers and cam timers.
PLCs are useful tools for repeatable processes because they have no mechanical parts. Each CPU continually loops through an input scan, program scan, output scan and housekeeping mode, repeatedly performing a single task while monitoring conditions. The information the controller gathers provides feedback to guide changes and improvements to processes -- some of which can be performed automatically depending on the devices coding.
A PLC controls individual devices, while a DCS can control multiple machines in a plant. And, while PLCs are designed to replace mechanical devices like mechanical relays, DCSs are able to control entire factories and process plants with many interconnected systems.
Other industrial control system technologies include programmable automation controllers, industrial automation and control systems, remote terminal units, control servers, intelligent electronic devices and sensors.

Chapter-5

What is a SCADA System?

SCADA stands for **supervisory control and data acquisition**.

- Supervisory control and data acquisition (SCADA) is defined as a comprehensive hardware and software solution that controls and manages high-level industrial processes without human intervention.
- SCADA works by gathering real-time data remotely to process it and control conditions and equipment. Enterprises also leverage SCADA to make data-driven decisions regarding industrial processes.

SCADA Components

The SCADA system components include the following.

Supervisory System or MTU (Master Terminal Unit)

The supervisory controller, also called the master terminal unit (MTU), plays the role of a central communication server. It is located in the control center and manages information exchange between the human-machine interface and the RTUs, sensors, PLCs, and other devices. A single personal computer could satisfactorily serve as the supervisory controller in a smaller SCADA configuration. However, the master terminal unit for larger SCADA deployments typically includes numerous servers, distributed software applications, and measures for disaster recovery. Such a system may use hot-standby measures to ensure that critical industrial processes remain unaffected in case of a system failure.

RTUs (Remote Terminal Units)

The RTU or remote terminal unit is an electronic device and it is also known as remote telemetry units. This system comprises physical objects that are interfaced through RTUs. The controlling of these devices can be done through microprocessors. Here, microprocessors are utilized for controlling RTUs which are used to transmit the recorded data toward the supervisory system. The data can be received from the master system for controlling the connected objects.

PLCs (Programmable Logic Controllers)

The term PLC stands for programmable logic controllers which are used in SCADA systems with the help of sensors. These controllers are connected to the sensors for converting the output signal of the sensor into digital data. As compared with RTUs, these are used due to their flexibility, configuration, versatility & affordability.

Communication Infrastructure

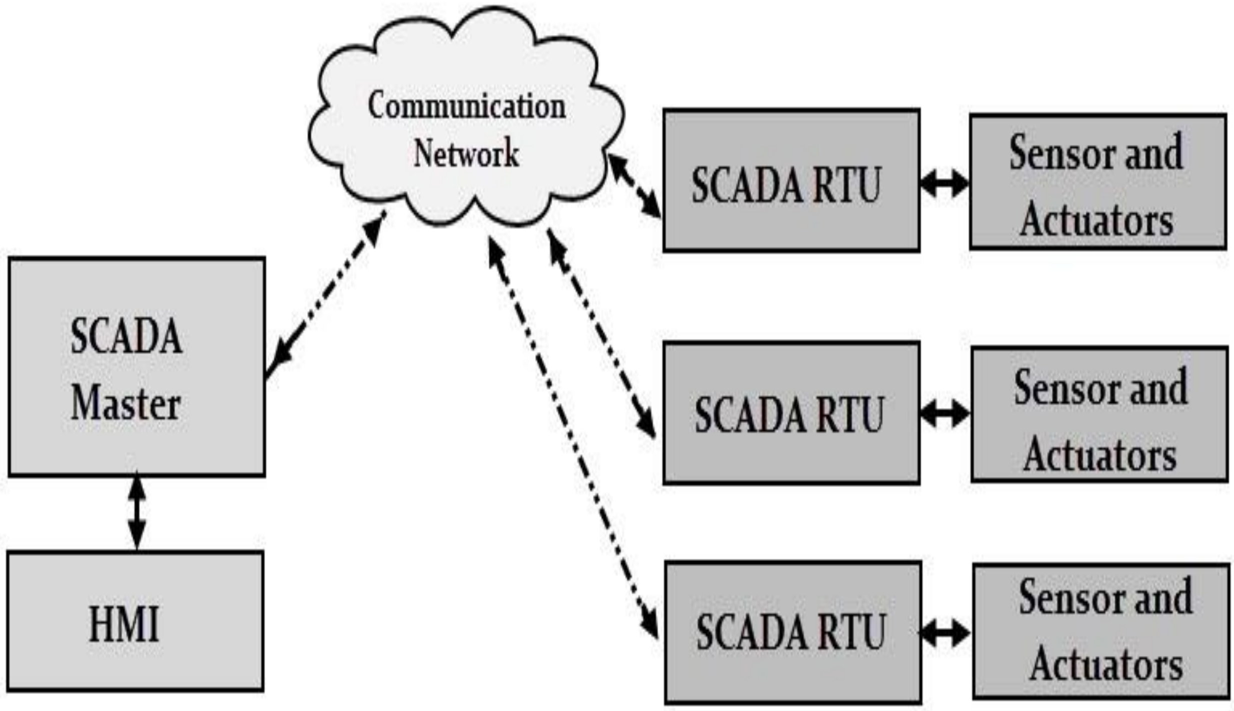
In the SCADA system, a mix of radio & the direct-wired connection is used. But, SONET or SDH can also be utilized for superior systems such as power stations & railways. Few standardized 7 recognized protocols are used between the compact SCADA protocols to deliver information simply once the RTUs are polled through the supervisory station.

SCADA Programming

In HMI otherwise master station, SCADA programming is mainly used to make maps, diagrams to provide very important information throughout progression otherwise when event failure occurs. Most of the commercial SCADA systems utilize consistent interfaces in C programming language otherwise derived programming language can also be used.

Human Machine Interface

The SCADA system uses the human-machine interface. The information is displayed and monitored to be processed by a human. HMI provides access to multiple control units which can be PLCs and RTUs. The HMI provides the graphical presentation of the system. For example, it provides a graphical picture of the pump connected to the tank. The user can see the flow of the water and the pressure of the water. The important part of the HMI is an alarm system that is activated according to the predefined values.



Difference between PLC and SCADA

The difference between PLC and SCADA include the following.

PLC	SCADA
The term PLC stands for programmable logic control	The term SCADA stands for Supervisory Control and Data Acquisition
PLC is hardware-based	SCADA is software-based
PLCs are mainly used to control the process of complex industries like motors and running machines.	SCADA is used to observe & run the processes of the plant.

The PLC includes Processor, I/O Modules, a Programming Device & Power Supply	The SCADA system includes three essential components like MTU, RTU, and HMI
There are different types of PLC like fixed or compact & modular.	The different types of a SCADA system are monolithic, distributed, networked & IoT
The i/p & o/ps are signified in NO (normal open), NC (normal close) & coil contacts.	The input & outputs of SCADA are represented through images.
In PLC, every component can be defined through an address.	In SCADA, each component can be defined through the name.

Advantages

The advantages of the SCADA system include the following:-

- The quality of service can be improved
- Reliability can be improved
- Maintenance cost is less
- The operation can be reduced
- Large system parameters can be monitored
- Manpower can be reduced
- Repair time can be reduced
- Fault detection & fault localization
- It stores a large amount of data
- As per the user requirement, it displays the data in various formats.
- Thousands of sensors can be interfaced with SCADA for controlling and monitoring
- Real data simulations can be obtained by operators
- Gives fast response
- It is flexible as well as scalable while adding extra resources.
- The SCADA system provides onboard mechanical and graphical information
- The SCADA system is easily expandable. We can add a set of control units and sensors according to the requirement.
- The SCADA system is able to operate in critical situations.

Disadvantages

The disadvantages of the SCADA system include the following:-

- It is complex in terms of dependent modules & hardware units.
- It needs analysts, programmers & skilled operators to maintain
- High installation cost
- Unemployment rates can be increased
- This system supports hardware devices and restricted software's

Applications

The applications of the SCADA system include the following:-

- Generation and Distribution of Power
- Public Transport

- Water and Sewage System
- Manufacturing
- Industries & Buildings
- Communication Networks
- Oil & Gas Industries
- Power generation, transmission, and distribution
- Water distribution and reservoir system
- Public buildings like electrical heating and cooling system.
- Generators and turbines
- Traffic light control system

Characteristic	PLC	DCS
Market Introduction	1960's Gould Modicon	1975 Honeywell
Replacement of…	Electromechanical Relays	Pneumatic & Single Loop Controllers
Products Manufactured…	"Things"	"Stuff"
Classic Application	Automotive	Refining
Type of Control	Discrete	Regulatory
Typical Configuration Languages	Ladder Logic	Function Block
Logic Execution Scan Time	20 ms and less	100 ms – 1 sec
Size / Footprint	Compact	Large

PLC	SCADA	DCS
The full form of PLC is "Programmable Logic Controller"	The full form of SCADA is "Supervisory Control and DATA Acquisition"	The full form of DCS is "Distributed Control System"
A centralized controller is used for automation tasks	Pics of software that integrate with multiple PLC and RTU for controlling and monitoring purposes.	Dedicated solution providing for large and complex process industries.
Software used for programming provided by Manufacturer companies	Multiple software vendors are available in the market.	Software is provided only by DCS Vendors company
Redundancy is possible	Redundancy is possible	Redundancy is possible at every level
Downtime is only converging into loss of production	Downtime is converging into loss of production, equipment failure, and creating hazardous situations	Downtime is very crucial because massive dangerous situations will be created and converge into losses of money and lives
Basic control strategies are implemented	Complex control strategies are implemented	Complex and more advanced control strategies are implemented
Basic communication structure is used	Advance and foster communication structures are used	Advance and complex communication is used with high-speed communication.
Implications const is low compared to other	Implementation cost is higher than PLC but lower than DCS	The cost of DCS implementation is the highest
PLC provider companies: – Siemens, ABB, Allan Bradley, Schneider	SCADA software provider companies: – Inductive automation, Rockwell Automation, Siemens, Aveva	DCS providers companies: – Yokogawa, ABB, Emerson, Rockwell Automation, Schneider

- Small Packaging industries
- CNC machines
- Small metal cutting industry
- Injection modeling machine

- Automobile etc.
- Food processing
- Dairy Industries

- Oil and Gas Refinery
- Power plant
- Water Treatment Plant
- Cement making industries
- Some Chemical Plants

Features of SCADA:

- Data acquisition is done by the Master Station with the help of RTUs
- Display of information in the form of pictures or text is provided on several HMIs
- The SCADA executes supervisory form of control. Control of equipment which are at remote locations is done from the master station
- Alarm Processing – There is facility to alert the operator by informing the place and time of an event.
- Information storage and reports – Data is stored in a temporary data base for 40 days or 12 months. Later it is shifted to a permanent storage device.

Chapter-6

Drive

Systems employed for motion control are called drives and may employ any of the prime movers. Drives employing electric motors are known as electric drives.

AC drive

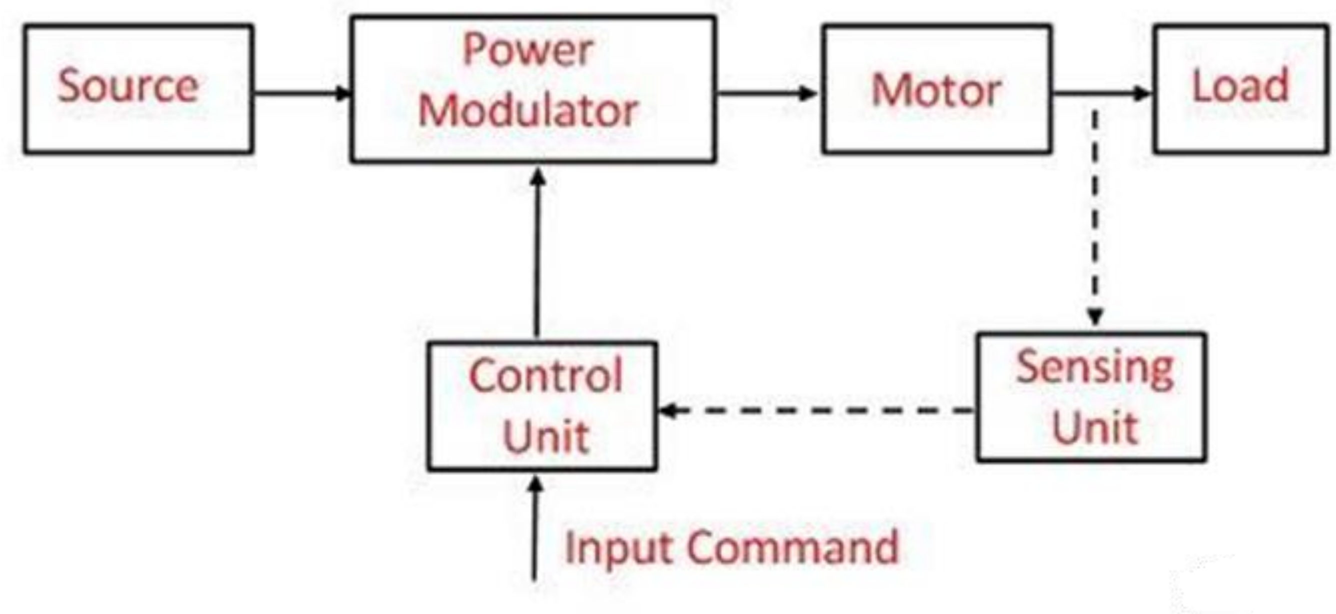
An AC drive is a device used to control the speed of an electrical motor.

Electrical Drive

The drive which uses the electric motor is called electrical drive. The electrical drive uses any of the prime movers like diesel or a petrol engine, gas or steam turbines, steam engines, hydraulic motors and electrical motors as a primary source of energy. This prime mover supplies the mechanical energy to the drive for motion control.

Electrical Drive Systems

The electrical drive system is defined as the system which is use for controlling the speed, torque and direction of an electrical motor. Each electrical drive system is different from other electrical drive systems, but there are some common features associated with all electrical drive systems.

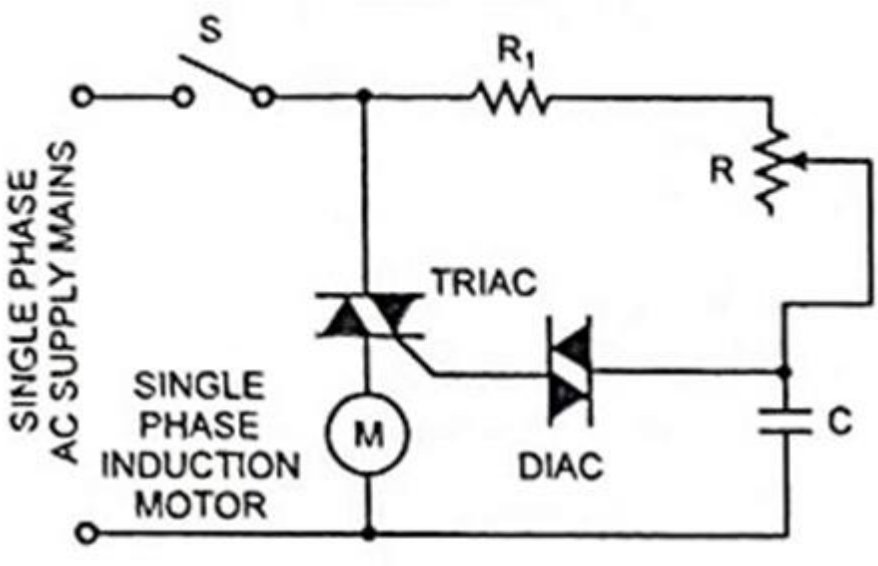


Block diagram of electric drive System

Speed Control of Single Phase Induction Motor using Triac

The conventional speed regulator for a fan uses a resistance regulator. The regulator resistance is in series with the fan motor. The speed of the fan is reduced, whenever desired, by increasing the regulator resistance. Thus the voltage drop across the regulator resistance increases and, therefore, voltage applied to the motor is reduced and so the speed. The resistance regulator causes loss of energy. This loss of energy becomes significant at low speeds.

Working Principle:



Speed Control of Single Phase Induction Motor using Triac

When voltage across capacitor C exceeds breakdown voltage of diac, it is triggered into conduction and sends a triggering signal to the triac gate. When triac is turned on the motor starts.

By varying resistance R, The firing angle of triac can be altered and thus the voltage applied to the motor is changed and consequently the motor speed is changed. Since resistance R, carries very small current, the loss of energy is very small. Sometimes an R-C snubber circuit is added in parallel with the triac to protect it from high voltage transients and high dv/dt.

Advantages of Electrical Drive

- The electric drive has very large range of torque, speed and power.
- Their working is independent of the environmental condition.
- The electric drives are free from pollution.
- The electric drives operate on all the quadrants of speed torque plane.
- The drive can easily be started and it does not require any refuelling.
- The efficiency of the drives is high because fewer losses occur on it.

Disadvantages of Electrical Drive

The power failure completely disabled the whole of the system.

1. The application of the drive is limited because it cannot use in a place where the power supply is not available.
2. It can cause noise pollution.
3. The initial cost of the system is high.
4. It has a poor dynamic response.
5. The output power obtained from the drive is low.

6. During the breakdown of conductors or short circuit, the system may get damaged due to which several problems occur.

Application of Electric Drive

It is used in a large number of industrial and domestic applications like transportation systems, rolling mills, paper machines, textile mills, machine tools, fans, pumps, robots and washing, etc.

What is a VFD (Variable Frequency Drive) and where is it used?

VFD is the short term for Variable Frequency Drive. A VFD or Variable Frequency Drive is also often commonly known as Frequency Inverter, Frequency Converter or AC drive.

The VFD is a motor control device that protects and controls the speed of an AC induction motor. The VFD controls the speed of the motor during the start and stop cycle, as well as the entire run cycle.

How does a VFD (Variable Frequency Drive) work?

A VFD (Variable Frequency Drive) converts input AC power to adjustable frequency and voltage source for controlling speed of the AC induction or PM motor. The frequency of the power applied to an AC motor determines the motor speed, based on the following:

$N = 120 \times f / p$ = speed (rpm)

Where f = frequency (Hz) p = number of motor poles

Working principle of the VFD

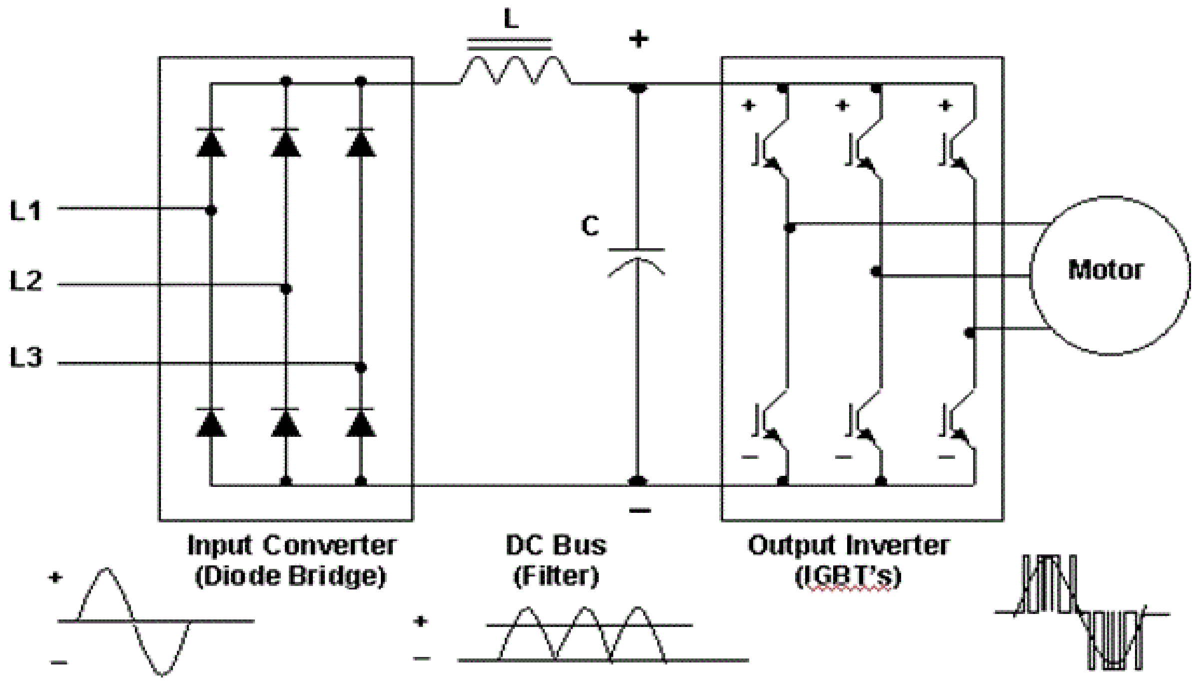
AC voltage is transformed into DC voltage. This DC voltage is sent out in pulses, both positively and negatively and generates a sinusoidal-like motor current. The frequency of this "sine wave" can be adjusted by varying the alternating positive and negative DC voltages per time unit. It is possible to control the height of the output voltage with the width of the pulsating DC voltage.

alternating voltage -> direct voltage -> adjustable alternating voltage

AC -> DC -> adj AC

Transforming DC voltage into AC voltage

The VFD then transforms DC voltage into an AC voltage. The VFD can be represented as six "switches". Each switch interrupts the DC current during some time, thus allowing the generation of a voltage approaching the sinusoidal.



VFD Switches

There are several different components used as switches in the VFD; thyristors, transistors, MOSFET (Metal Oxide Silicon Field Effect Transistor) and IGBT (Insulated Gate Bipolar Transistor). Today mainly IGBTs are used.

Where can I use a VFD?

- *VFD for pumps and compressors*
- *VFD for fans and blowers*
- *VFD for cranes, lifts and conveyor belts*
- *VFD for crushers, mixers, mills, band saws*
- *VFD for propulsion, shaft power generation, thrusters and other marine applications*

Comparison between AC and DC Drives.

Characteristics	AC Drives	DC Drives
Definition	AC drives (also known as VFD) convert the AC supply to DC using converter (rectifier) and invert it back from the DC to the AC using	DC drives only convert the input AC supply to the DC using converter circuit based on rectifier to run the DC

	inverter to run the AC motors.	motors.
Control	AC drives control AC output from AC input.	DC drives control DC output from AC input.
Main Supply & Voltage	AC Drives run by AC power supply i.e. single phase and three phase AC voltages.	DC Drives run by DC power supply i.e. Batteries and supplies sources of DC voltages.
Self Start	Not self starting	Self Starting
Circuit Design	The circuit design of AC drives is little bit complex due to the inverter and converter which convert AC to DC and invert back DC to AC then.	The circuit design of DC drives is less complex due to the single power conversion i.e. It converts AC to DC only once.
Power and Control Circuits	AC drives power and control circuits are complex in design as compared to the DC drives.	DC drives power and control circuits are simple in design as compared to the AC drives.
Breaking / Acceleration	The breaking and acceleration mechanism of AC drives can be controlled by changing the supply frequency (F_s).	The breaking mechanism of DC drives can be control by applying resistance at rotor side.
Speed Control	Speed control is done by changing the supply frequency.	Speed control is done by armature and field control.
Speed Limitation	Maximum speed can be achieved.	The speed is limited because of commutator used in the motors for commutation.
Motor Speed in RPM	Up to 10k RPM	Up to 2.5k RPM
Speed Regulation	~1% speed regulation is achievable in motors run by AC drives.	1% speed regulation is not possible to achieve in motors run by DC drives.
Speed Torque	It is complex to adjust the speed torque curves.	It can be achieved easily.

Starting Torque	Low	High
Inverter / Converter	AC drives have both inverter and converter.	DC drives have converter and chopper circuits. It doesn’t need an inverter.
Commutator and Slip Rings	No need of commutation but slip rings in AC drives.	No slip rings but Commutation is needed in DC drives.
Rectification	No need of rectification circuit.	Rectifier circuit is must.
Battery Operation	AC drives won’t be connected directly to the batteries as some additional components and circuits are needed to do so.	DC drives can be connected and run directly through batteries (providing DC voltage).
Transformer	AC drives can be directly connected to the transformer (mains supply).	In DC drives, transformer is needed at voltage higher than 100V.
Power usage	AC drives consume less power as compared to DC drives.	DC drives consume more power as compared to AC drives.
Brushes Life span	High (about 10k hours)	Low (about 3k hours)
Noise	Operation of AC drives is noisy.	Operation of DC drives is less noisy.
Harmonics	Inverter used in AC drives produces harmonics on both side i.e. supply and load.	Converter used in DC drives doesn’t generate harmonics.
Spark	AC drives are spark free and can be used in wet environment.	DC drives can’t be used in wet areas due to generated spark in the burses.
Dynamic Response	High	Low
Maintenance	Less	More and frequent
Size, Weight & Power Rating	Large	Small

Cost	AC drives are more expensive while the motors used in AC drives are less expensive e.g. squirrel cage motor.	DC drives are less expensive while the motors used in DC drives are little bit expensive.
General uses	AC drives used in almost all areas with vast applications.	DC drives are used in few areas as compared to AC drives.
Applications	Generally, AC drives used for AC motors. They are used to control the speed of AC motors.	Normally, DC drives used for DC motors. They are used to control the speed of DC motors.