

Instrumentation and Measurement

**Prepared by
Sakibul Islam Swan
Lecturer
Department of Mechanical Engineering
University of Global Village**

Basic Course Information

Course Title	Instrumentation & Measurement
Course Code	ME-361
Credits	03
CIE Marks	90
SEE Marks	60
Exam Hours	2 hours (Mid Exam) 3 hours (Semester Final Exam)
Level	6th Semester
Academic Session	Winter 2025

ASSESSMENT PATTERN
CIE- Continuous Internal Evaluation (90 Marks)

Bloom's Category Marks (out of 90)	Tests (45)	Assignments (10)	Class Test (20)	Quiz (5)	External Participation in Curricular/Co-Curricular Activities (10)
Remember	5		10	05	
Understand	5	05	10		
Apply	10				10
Analyze	15				
Evaluate	10				
Create		05			

SEE- Semester End Examination (60 Marks)

Bloom's Category	Test
Remember	10
Understand	10
Apply	10
Analyze	10
Evaluate	10
Create	10

Course Learning Outcomes

- **CLO-1:** Develop a foundational understanding of measurement principles and practices.
- **CLO-2:** Demonstrate proficiency in selecting, applying, and analyzing various types of sensors and transducers.
- **CLO-3:** Acquire the ability to design and implement effective signal conditioning and data acquisition systems.
- **CLO-4:** Gain the expertise to design, implement, and troubleshoot complete instrumentation systems for a range of applications.

Course Objectives

The objectives of an Instrumentation and Measurement course are to:

- Introduce fundamental measurement principles and techniques.
- Explore various sensors and transducers for different physical quantities.
- Teach signal conditioning and data acquisition methods.
- Develop skills in designing and implementing instrumentation systems.
- Provide practical experience with laboratory instruments.
- Foster problem-solving and critical thinking in measurement applications

Course Summary

Serial No	Course Content	Hours
01.	Functional elements of a measurement system; Sensors for temperature, flow, velocity, linear distance, level, pressure, force, pH measurement; Control valve sizing and selection	5
02.	Control signal transmission and related industry standards; Study of piping instrument diagram (P&ID); Basic components of Data acquisition system, Error analysis and calibrations.	5
03.	Introduction of PLC, PLC Hardware, Definitions of Allen-Bradley conditional inputs and outputs, I/O configuration, Programming Terminals and Peripherals, Application of PLC in Automation,	15
04.	Automation system components, Industrial communications, Continuous control, overview of SCADA and DCS systems; Microcontroller basics, classification, basic Architecture, memory, registers, I/O ports	15

Course Summary

Serial No	Course Content	Hours
05.	Basics of feedback control; P, PI, and PID controller design, tuning, and troubleshooting	5
06.	Frequency response analysis and control system design, Stability analysis; Basics of cascade control; fuzzy logic, control structure	5

Course Plan Mapped with CLO

Week No.	Topics	Teaching Learning Strategy	Assessment strategy	Alignment To CLO
1.	Functional elements of a measurement system; Sensors for temperature, flow, velocity, linear distance, level, pressure, force, pH measurement	Lecture, Multimedia	Feedback, Q&A	CLO 1
2.	Control valve sizing and selection, Control signal transmission and related industry standards;	Lecture, Discussion Multimedia	Feedback, Q&A	CLO 1
3.	Study of piping instrument diagram (P&ID); .	Lecture, Multimedia	Feedback, Q&A	CLO 2
4.	Basic components of Data acquisition system, Error analysis and calibrations.	Lecture, Multimedia	Feedback, Q&A	CLO 2

Course Plan Mapped with CLO

Week No.	Topics	Teaching Learning Strategy	Assessment strategy	Alignment To CLO
5.	Introduction of PLC, PLC Hardware, Definitions of Allen-Bradley conditional inputs and outputs	Lecture, Multimedia	Feedback, Q&A	CLO 2
6.	I/O configuration, Programming Terminals and Peripherals,	Lecture, Discussion Multimedia	Feedback, Q&A	CLO 2
7.	Application of PLC in Automation,	Lecture, Multimedia	Feedback, Q&A	CLO 3
8.	Automation system components, Industrial communications, Continuous control,	Lecture, Multimedia	Feedback, Q&A	CLO 3

Course Plan Mapped with CLO

Week No.	Topics	Teaching Learning Strategy	Assessment strategy	Alignment To CLO
9.	overview of SCADA and DCS systems	Lecture, Multimedia	Feedback, Q&A	CLO 3
10.	Microcontroller basics, classification, basic Architecture, memory, registers, I/O ports	Lecture, Discussion Multimedia	Feedback, Q&A	CLO 3
11.	Microcontroller basics, classification, basic Architecture, memory, registers, I/O ports	Lecture, Multimedia	Feedback, Q&A	CLO 4
12.	Basics of feedback control;	Lecture, Multimedia	Feedback, Q&A	CLO 4

Course Plan Mapped with CLO

Week No.	Topics	Teaching Learning Strategy	Assessment strategy	Alignment To CLO
13.	P, PI, and PID controller design, tuning, and troubleshooting	Lecture, Multimedia	Feedback, Q&A	CLO 4
14.	Frequency response analysis and control system design	Lecture, Discussion Multimedia	Feedback, Q&A	CLO 4
15.	Stability analysis;	Lecture, Multimedia	Feedback, Q&A	CLO 4
16.	Basics of cascade control; fuzzy logic, control structure	Lecture, Multimedia	Feedback, Q&A	CLO 4

Reference Books

1. Process Dynamics and Control, 4th Ed. – Dale.E. Seborg, Thomas F. Edgar et al.
2. Process Control, 2nd Ed. – Thomas E. Marlin
3. Schaum's Outline of Feedback and Control Systems, 2nd Ed. - Allen J. Stubberud
Ivan J. Williams Joseph J. DiStefano
4. Programmable Logic Controllers: Principles and Applications, 5th Ed.-John W.
Webb, Ronald A. Reis
5. Programming and Customizing the AVR Microcontroller-Gadre, McGraw-Hill

Instrumentation:

Instrumentations defined as the art and science of measurement and control of the process variables within a production or manufacturing area.

The process variables used in industries are Level, Pressure, Temperature, Humidity, Flow, pH, Force, Speed etc. Instrumentation Engineering deals with the design, configuration and automation of systems.

Electronics & Instrumentation Engineering (E&I), Instrumentation & Controls Engineering (ICE), are the two Engineering courses which are derivative of the Instrumentation Engineering. These two courses covers almost similar subjects and the professionals graduated in these courses can look for a similar work profile.

What is the future for an Instrumentation graduate?

Instrumentation Engineers are responsible for the design, construction and maintenance of the entire instrumentation systems of any industrial undertaking.

The type of instruments needed for ensuring better quality and efficiency of the end product are decided by an instrumentation engineer.

Instrumentation Engineers design the actuators, positioning devices and robots for industrial automation.

Instrumentation and control engineers monitor measure and control physical quantities like temperature, pressure and flow in an automated industrial environment.

Job Prospects

- Instrumentation engineers can get jobs in R&D(Research and Development) units of public and private sector companies. They are also required by the Heavy industries such as Thermal Power Stations, Steel Plants, Refineries, and Cement and Fertilizer Plants. They have a multidisciplinary role to play. One may choose to move sideways into other career areas either within or outside their industry. This might include areas such as purchasing, sales, marketing, finance, HR, IT or general management.
- These engineers can pursue consultancy-based work. Those who have an aptitude use their expertise they have gained in industry and engage in academic research in universities or acquire a tutoring/coaching role as a lecturer or trainer of instrumentation engineers. Senior level positions are occupied with a Masters Degree and they carry the highest level of responsibility and may include planning and managing activities, as well as leading on new developments. Senior engineers in production and operation functions can often be representatives at board level.

Nature of Work

A control and instrumentation engineer is essentially responsible for designing, developing, installing, managing and/or maintaining equipment which is used to monitor and control engineering systems, machinery and processes.

Tasks and responsibilities, which are common to instrumentation engineers, may include:

- Designing and developing new control systems;
- Maintaining and modifying existing systems;
- Managing operations;
- Working collaboratively with design engineers, operation engineers, purchasers and other internal staff;
- Contacting clients, suppliers, contractors and relevant authorities
- Project management within cost and time constrained environments;
- Troubleshooting and problem-solving;
- Understanding and ensuring compliance with the health and safety regulations and quality standards
- Providing advice and consultancy support;
- Purchasing equipment;
- Writing computer software;

Control engineering

Control systems engineering is the engineering discipline that applies control theory to design systems with desired behaviors. Control engineers are responsible for the research, design, development and control devices/systems, typically in manufacturing facilities and plants.

The practice uses sensors to measure the output performance of the device being controlled and those measurements can be used to give feedback to the input actuators that can make corrections toward desired performance.

When a device is designed to perform without the need of human inputs for correction it is called automatic control (such as cruise control for regulating a car's speed). Multi-disciplinary in nature, control systems engineering activities focus on implementation of control systems mainly derived by mathematical modeling of systems of a diverse range.

Features of a Control System

The main feature of a control system is that there should be a clear mathematical relationship between input and output of the system. When the relation between input and output of the system can be represented by a linear proportionality, the system is called a linear control system. Again when the relationship between input and output cannot be represented by single linear proportionality, rather the input and output are related by some non-linear relation, the system is referred to as a non-linear control system.

Requirements of a Good Control System

- **Accuracy:** Accuracy is the measurement tolerance of the instrument and defines the limits of the errors made when the instrument is used in normal operating conditions. Accuracy can be improved by using feedback elements. To increase the accuracy of any control system error detector should be present in the control system.
- **Sensitivity:** The parameters of a control system are always changing with the change in surrounding conditions, internal disturbance or any other parameters. This change can be expressed in terms of sensitivity. Any control system should be insensitive to such parameters but sensitive to input signals only.
- **Noise:** An undesired input signal is known as noise. A good control system should be able to reduce the noise effect for better performance.
- **Stability:** It is an important characteristic of the control system. For the bounded input signal, the output must be bounded and if the input is zero then output must be zero then such a control system is said to be a stable system.
- **Bandwidth:** An operating frequency range decides the bandwidth of the control system. Bandwidth should be as large as possible for the frequency response of good control system.
- **Speed:** It is the time taken by the control system to achieve its stable output. A good control system possesses high speed. The transient period for such system is very small.
- **Oscillation:** A small numbers of oscillation or constant oscillation of output tend to indicate the system to be stable.

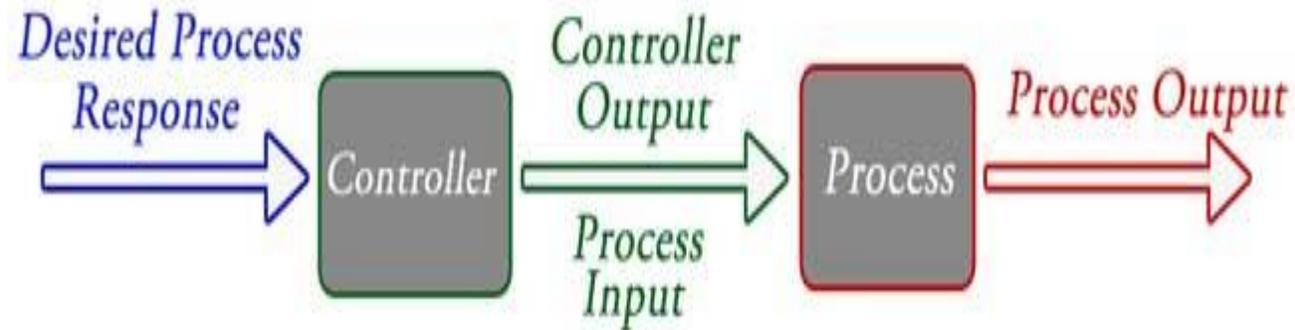
Hence, there are two main **types of control system**. They are as follow

1. Open loop control system

1. Closed loop control system

Open Loop Control System

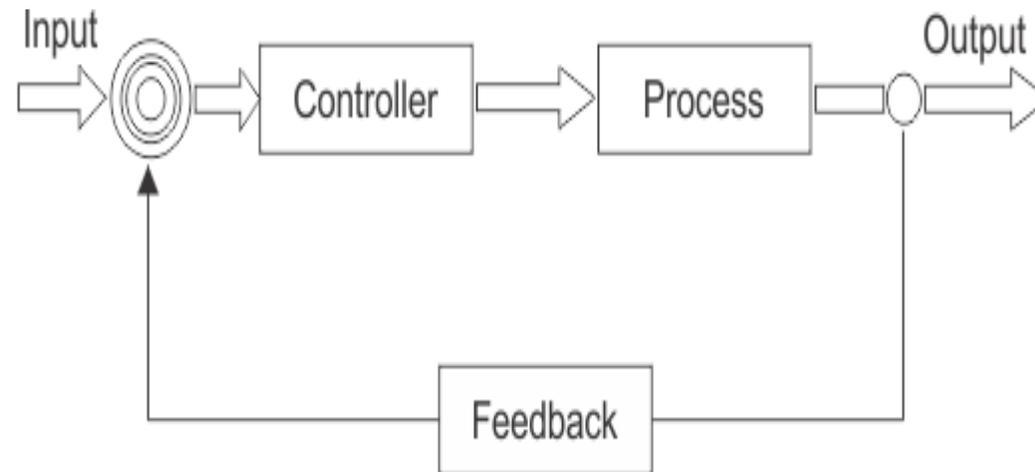
A control system in which the control action is totally independent of output of the system then it is called open loop control system. A manual control system is also an open loop control system. The figure below shows a control system block diagram of an open loop control system in which process output is totally independent of the controller action.



Closed Loop Control System

closed loop control system: Control system in which the output has an effect on the input quantity in such a manner that the input quantity will adjust itself based on the output generated is called **closed loop control system**.

Open loop control system can be converted in to closed loop control system by providing a feedback. This feedback automatically makes the suitable changes in the output due to external disturbance. In this way closed loop control system is called automatic control system. Figure below shows the block diagram of closed loop control system in which feedback is taken from output and fed in to input.

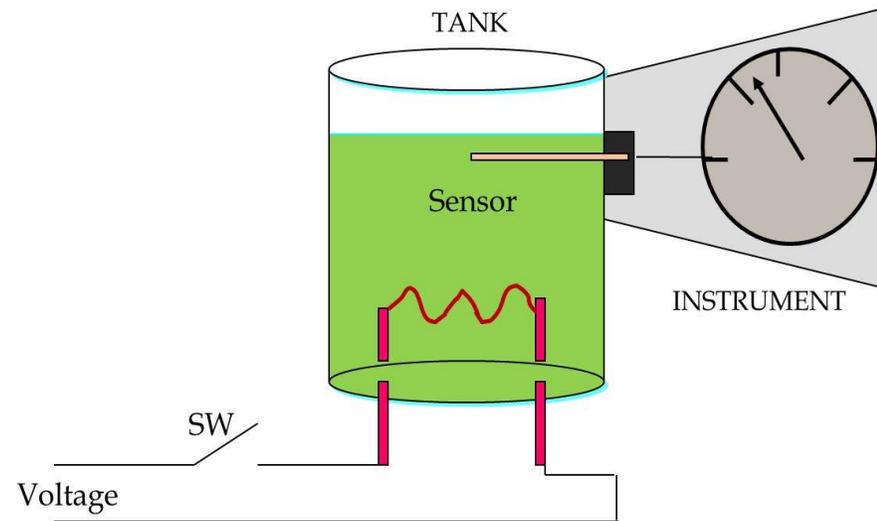


Instrumentation and control engineers work with the goal of improving

- Productivity,
- Optimization,
- Stability, dedication
- Reliability,
- Safety,
- Continuity.

What is a measurement instrument?

A measurement instrument is a device capable of detecting change, physical or otherwise, in a particular process. It then converts these physical changes into some form of information understandable by the user.

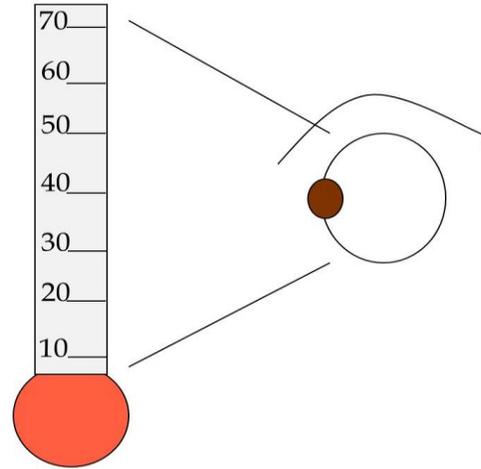


When the switch is closed, the resistor generates heat, increasing the temperature of the liquid in the tank. This increase is detected by the measurement instrument and shown on the scale of that instrument.

We can get the information on the physical changes in a process using direct indication or a recorder.

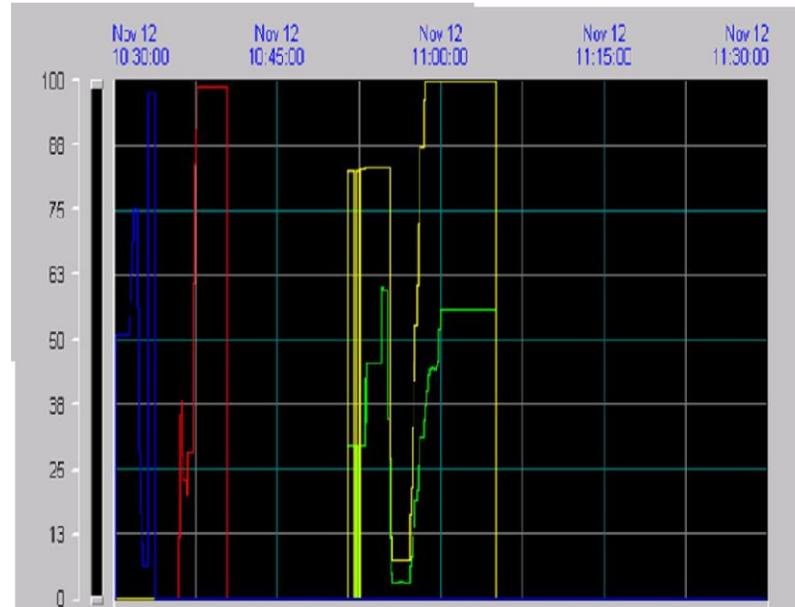
Indication

This is the simplest form of measurement; it allows us to know the current state of the variable.



Recorder

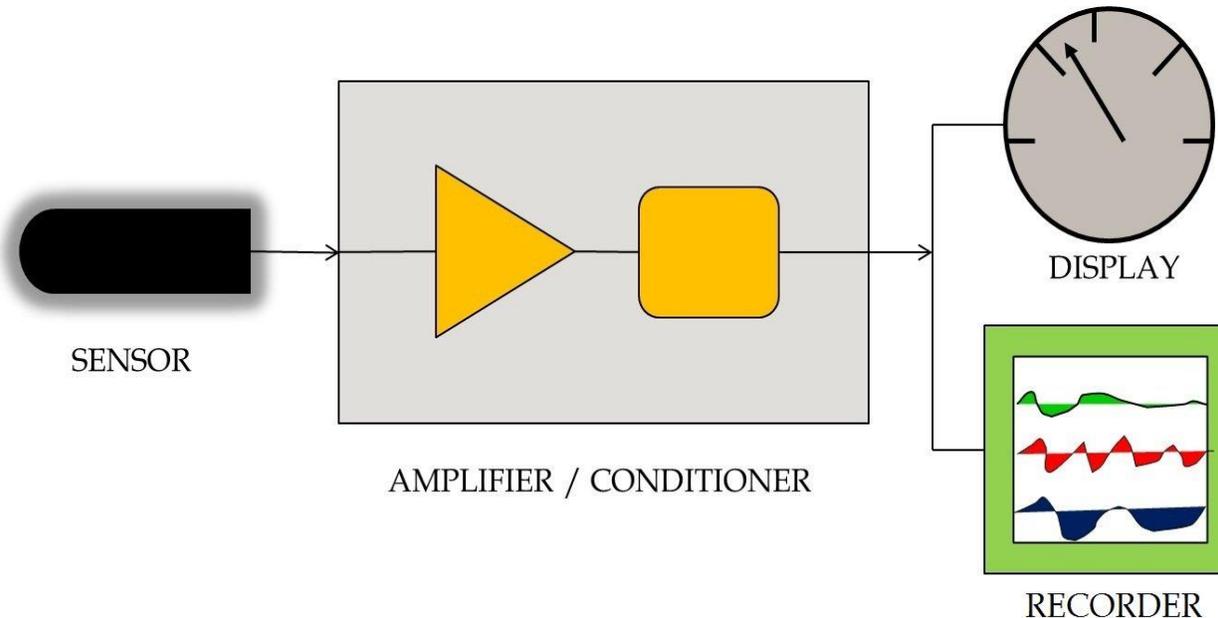
A device that can store data allows us to observe the current state of the variable and how it behaved in the past. A recorder provides us with the history of the variable.

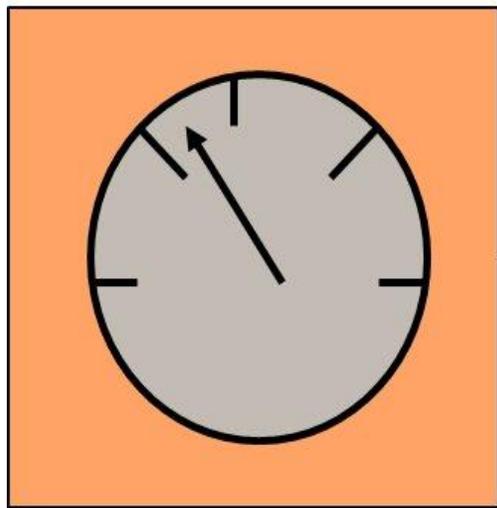


Elements of a Measurement Instrument

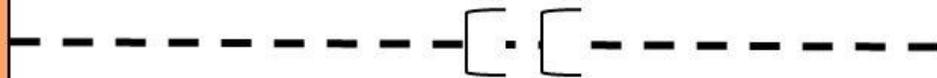
Measurement instruments consist primarily of the following parts:

- **Sensor:** This element is a device that experiences changes in its physical properties as a result of changes in the process it's measuring.
- **Amplifier / Conditioner:** Changes detected by the sensor may be very small, so they must be amplified and then conditioned such that they can be properly displayed.
- **Display:** The measured data should be presented in an understandable way. This can be done using a graduated instrument or an electronic display. Sometimes the display additionally acts as a recorder in order to convey the measurement's history or trends.

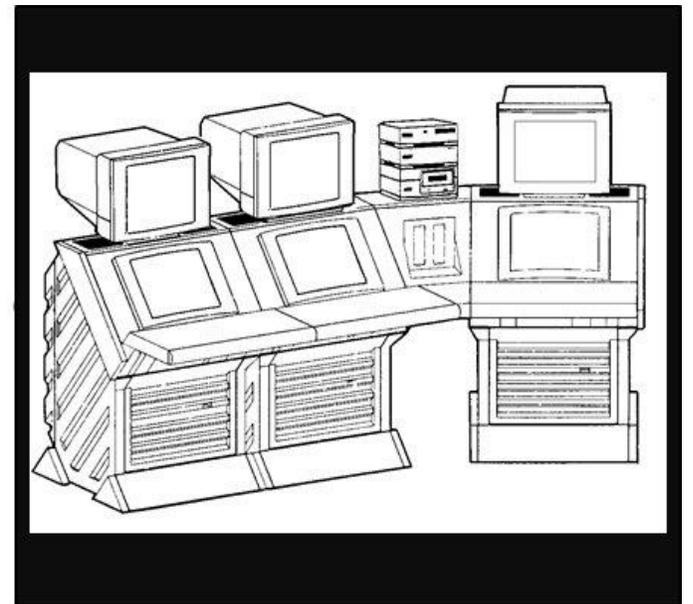




INSTRUMENT



$D = 1000 \text{ m}$



CONTROL ROOM

Classification of Instruments

There are different classifications for measurement instruments.

We can classify them, for example, as in-field instruments or panel instruments.

The in-field instrument is installed close to the process or measuring point. It must be physically robust if it will be exposed to harsh environmental conditions.

Panel instruments are in a controlled-environment room (often a clean space with air conditioning and controlled humidity).

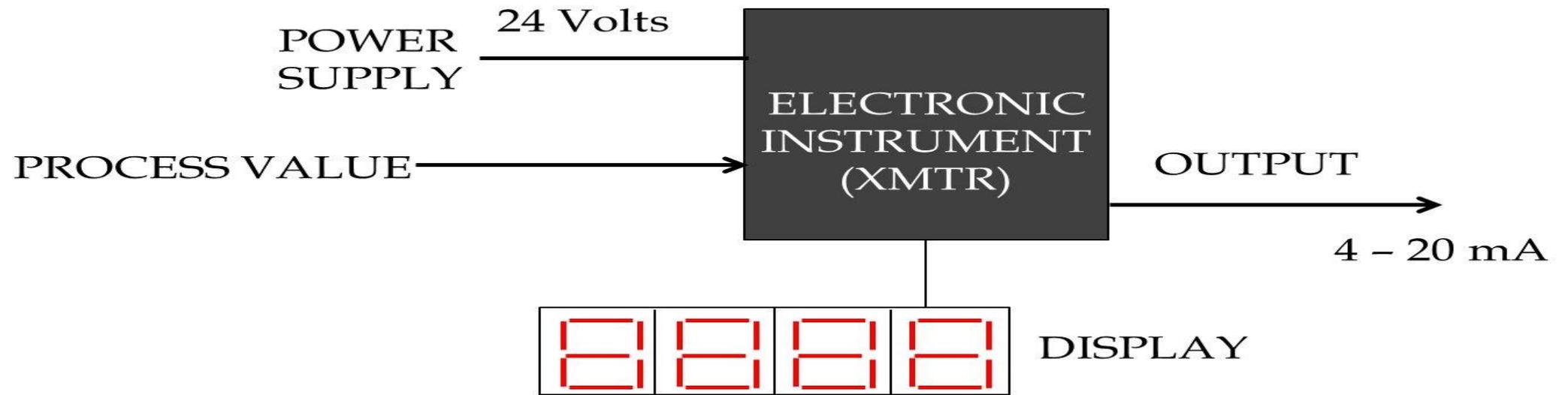
Another classification is pneumatic instruments vs. electrical/electronic instruments.

Electrical / Electronic Instruments

Electronic instruments can be divided into three general categories: analog, smart analog, and digital.

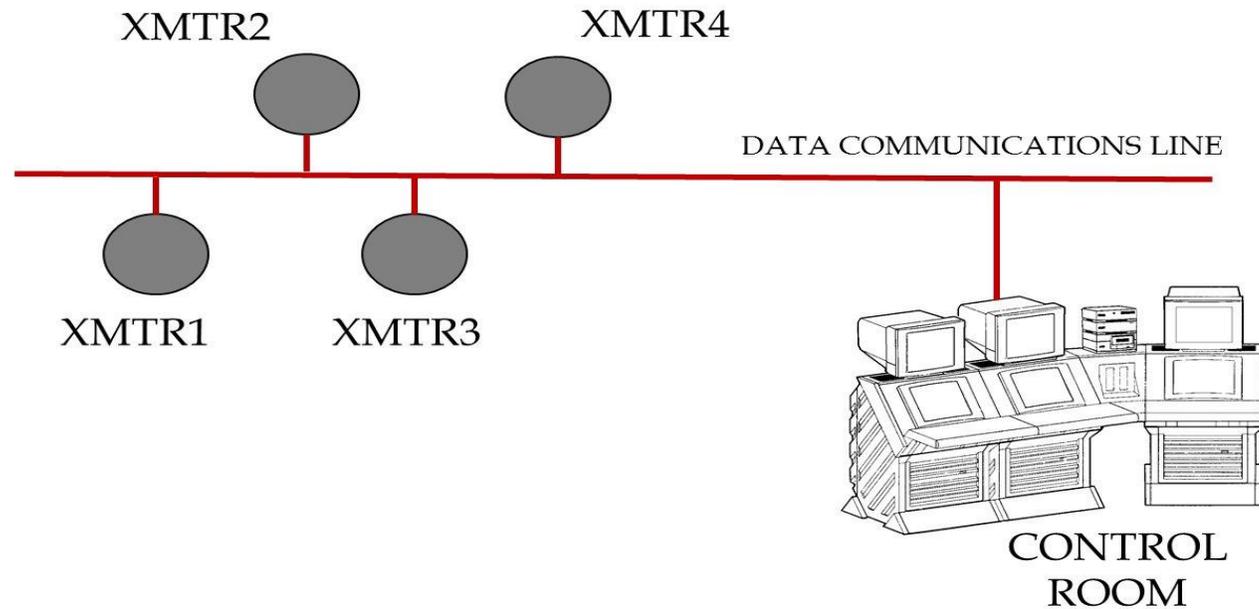
Analog:

- Output signal: 4 - 20 mA
- Transmission distance: 1200 m (typical)
- Data for one variable is transmitted
- Good accuracy
- Easy maintenance



Digital:

- Multiple instruments can use a single cable
- Transmission of multiple values for each instrument (process variables, calibration, diagnostics, range)
- Distance: approximately 1900 m without a repeater
- Data capacity is influenced by the mode of transmission (cable, fiber optic, wireless)



Smart Analog:

- Characterization of the sensor as measuring temperature, static pressure, etc.
- Excellent accuracy
- Self-diagnosis (i.e., the sensor can analyze problems in its own functionality)
- One variable

General Concepts

- **Range:** The region between the limits within which a variable is measured. It indicates the minimum and maximum values that limit the region. The range is expressed with two numbers, e.g., 10 to 20 °C, 10 to 150 V, 0 to 100%
- **Span:** Calculated as the maximum value of the range minus the minimum value of the range. Span is expressed with a single number in process units, e.g., 120 °C, 30 V, 150 liters per second.
- **Elevation:** If the lower limit of the range is a positive value, this lower limit is the elevation. Example: If the range is 50 °C to 200 °C, we can say that the elevation is 50 °C or 33.3% of the span.
- **Depression (also referred to as suppression):** If the lower limit of the range is negative, the absolute value of this lower limit is the depression. Example: If the range is -10 °C to 80 °C, we can say that the depression is 10 °C or 11.1% of the span.
- **Over range:** When a device is calibrated to operate within a certain range but may be subjected to values above or below that range, then it requires a protection mechanism to prevent damage to the instrument or to prevent the indicator from exceeding its upper or lower limit. When the measured values are above the maximum value, we have positive over range. When the measured values are below the minimum value, we have negative over range.

- **Error:** The difference between the measured value and the actual (or expected, or desired) value of a physical variable. The error can be positive or negative. When the measured value is greater than the actual value, the error is positive. When the measured value is less than the actual value, the error is negative.

If measured $>$ actual, error $>$ 0

If measured $<$ actual, error $<$ 0

The error can be expressed

in engineering units (e.g., °C, psi)

as a percentage of the span (e.g., +/- 3% of the span)

as a percentage of the measurement (e.g., +/- 5% of the measurement)

- **Reference value:** In a general sense, this refers to the actual, expected, or desired value of a variable. In the context of a feedback control system, the measured value is fed back and subtracted from the reference value in order to generate the error signal.
- **Accuracy:** A number that defines the limits of the error. When we say that an instrument has an accuracy of 0.1% of the span, this means that anywhere within the range, the readings do not differ from the actual value by more than 0.1% of the span.

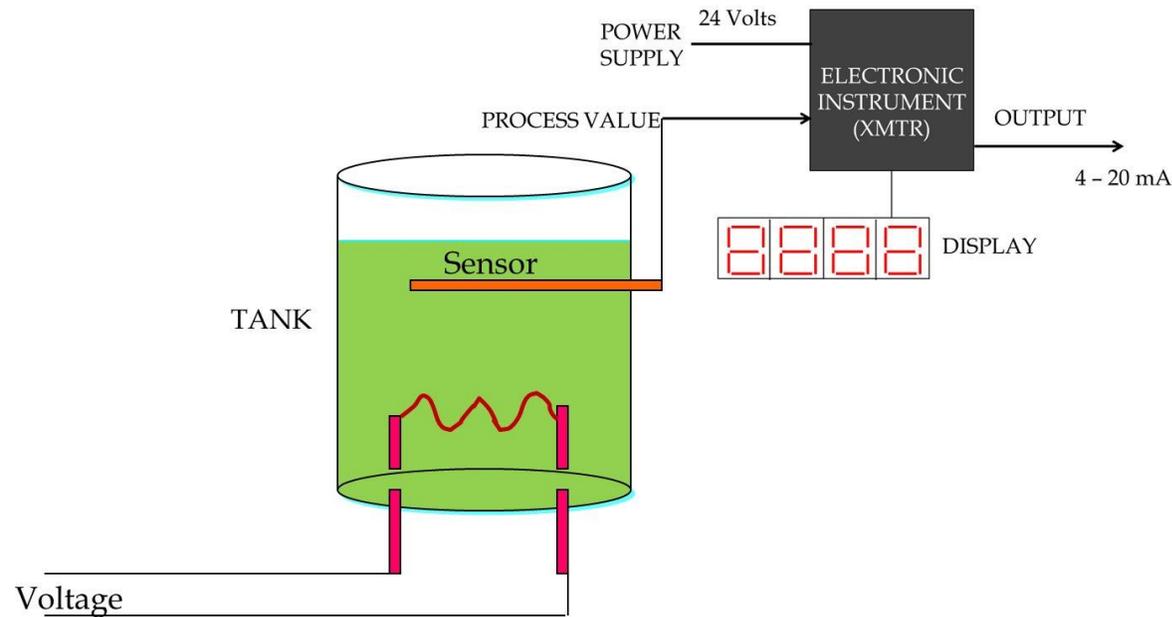
We have an oil tank where we are required to continuously measure the temperature. The operating conditions for this process are as follows:

Minimum temperature: $-10\text{ }^{\circ}\text{C}$

Maximum temperature: $90\text{ }^{\circ}\text{C}$

The measurement accuracy must be 1% of the span or better

The temperature measurement must be displayed locally and remotely



First, we must select a measuring instrument that allows us to measure the temperature of the liquid in the tank. Since the information should be available locally and remotely, we will choose a temperature transmitter.

This transmitter must have the following characteristics:

Range: $-10\text{ }^{\circ}\text{C}$ to $90\text{ }^{\circ}\text{C}$

Span: $90\text{ }^{\circ}\text{C} - (-10\text{ }^{\circ}\text{C}) = 100\text{ }^{\circ}\text{C}$

Depression: $10\text{ }^{\circ}\text{C}$ or 10% of the span

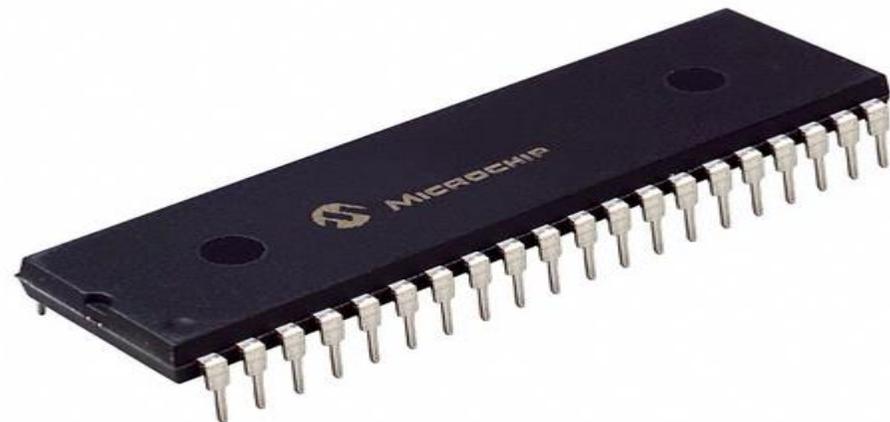
Accuracy: 1% of the span = $1\% \times 100\text{ }^{\circ}\text{C} = 1\text{ }^{\circ}\text{C}$

- This accuracy of 1% ensures that, in each measurement or temperature reading, variation or errors will not exceed $\pm 1\text{ }^{\circ}\text{C}$

On an additional note, we must ensure a proper relationship between the range and the standardized transmitter output. To calibrate the instrument, we must associate the minimum value of the range ($-10\text{ }^{\circ}\text{C}$) with the minimum value of the output (4 mA) and the maximum value of the range ($90\text{ }^{\circ}\text{C}$) with the maximum value of the output (20 mA).

What is a Microcontroller?

A microcontroller is a single chip microcomputer made through VLSI fabrication (Very-large-scale integration (VLSI) is the process of creating an integrated circuit (IC) by combining thousands of transistors into a single chip). A microcontroller also called an embedded controller because the microcontroller and its support circuits are often built into, or embedded in, the devices they control. A microcontroller is available in different word lengths like microprocessors (4bit,8bit,16bit,32bit,64bit and 128-bit microcontrollers are available today).



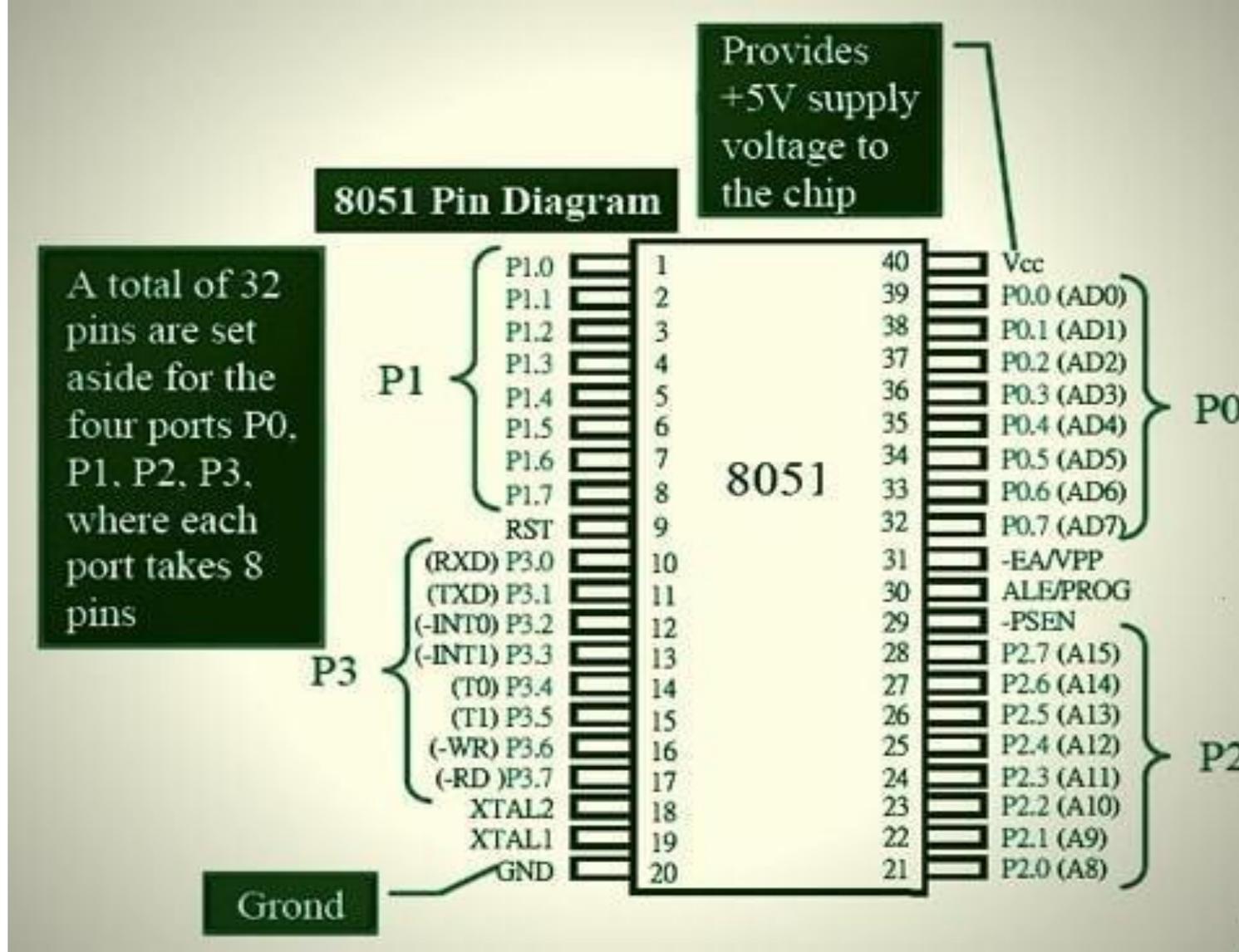
- ❖) A microcontroller internally consists of all features required for a computing system and functions as a computer without adding any external digital parts in it.
- ❖) Most of the pins in the microcontroller chip can be made programmable by the user.
- ❖) A microcontroller has many bit handling instructions that can be easily understood by the programmer.
- ❖) A microcontroller is capable of handling Boolean functions.
- ❖) Higher speed and performance.
- ❖) On-chip ROM structure in a microcontroller provides better firmware security.
- ❖) Easy to design with low cost and small size.

The Basics Of Microcontroller Programming

- ❑ write program code on your computer.
- ❑ compile the code with a compiler for the microcontroller you are using.
- ❑ upload the compiled version of your program to your microcontroller.

Which programming language is used for microcontrollers?

Microcontrollers were originally programmed only in **assembly language**, but various **high-level programming languages**, such as **C**, **Python** and JavaScript, are now also in common use to target microcontrollers and **embedded systems**. **One of the essential tools needed to program a microcontroller is an integrated development environment (IDE).**



Microcontrollers - 8051 Pin Description

- **Pins 1 to 8** – These pins are known as Port 1. This port doesn't serve any other functions. It is internally pulled up, **bi-directional I/O port**.
- **Pin 9** – It is a **RESET** pin, which is used to reset the microcontroller to its initial values.
- **Pins 10 to 17** – These pins are known as Port 3. This port serves some functions like interrupts, **timer input, control signals, serial communication** signals RxD and TxD, etc.
- **Pins 18 & 19** – These pins are used for interfacing an external crystal to get the system clock.
- **Pin 20** – This pin provides the **power supply** to the circuit.
- **Pins 21 to 28** – These pins are known as Port 2. It serves as **I/O port**. Higher order **address bus signals are also multiplexed using this port**.
- **Pin 29** – This is PSEN pin which stands for **Program Store Enable**. It is used to read a signal from the external program memory.
- **Pin 30** – This is EA pin which stands for **External Access** input. It is used to **enable/disable the external memory interfacing**.
- **Pin 31** – This is ALE pin which stands for **Address Latch Enable**. It is used to demultiplex the address-data signal of port.
- **Pins 32 to 39** – These pins are known as Port 0. It serves as I/O port. **Lower order address and data bus signals are multiplexed using this port**.
- **Pin 40** – This pin is used to provide power supply to the circuit.

How do microcontrollers work?

A microcontroller is embedded inside of a system to control a singular function in a device. It does this by **interpreting data it receives from its I/O** peripherals using its central processor. The temporary information that the microcontroller receives is stored in its data memory, where the processor accesses it and uses instructions stored in its program memory to decipher and apply the incoming data. It then uses its I/O peripherals to communicate and enact the appropriate action.

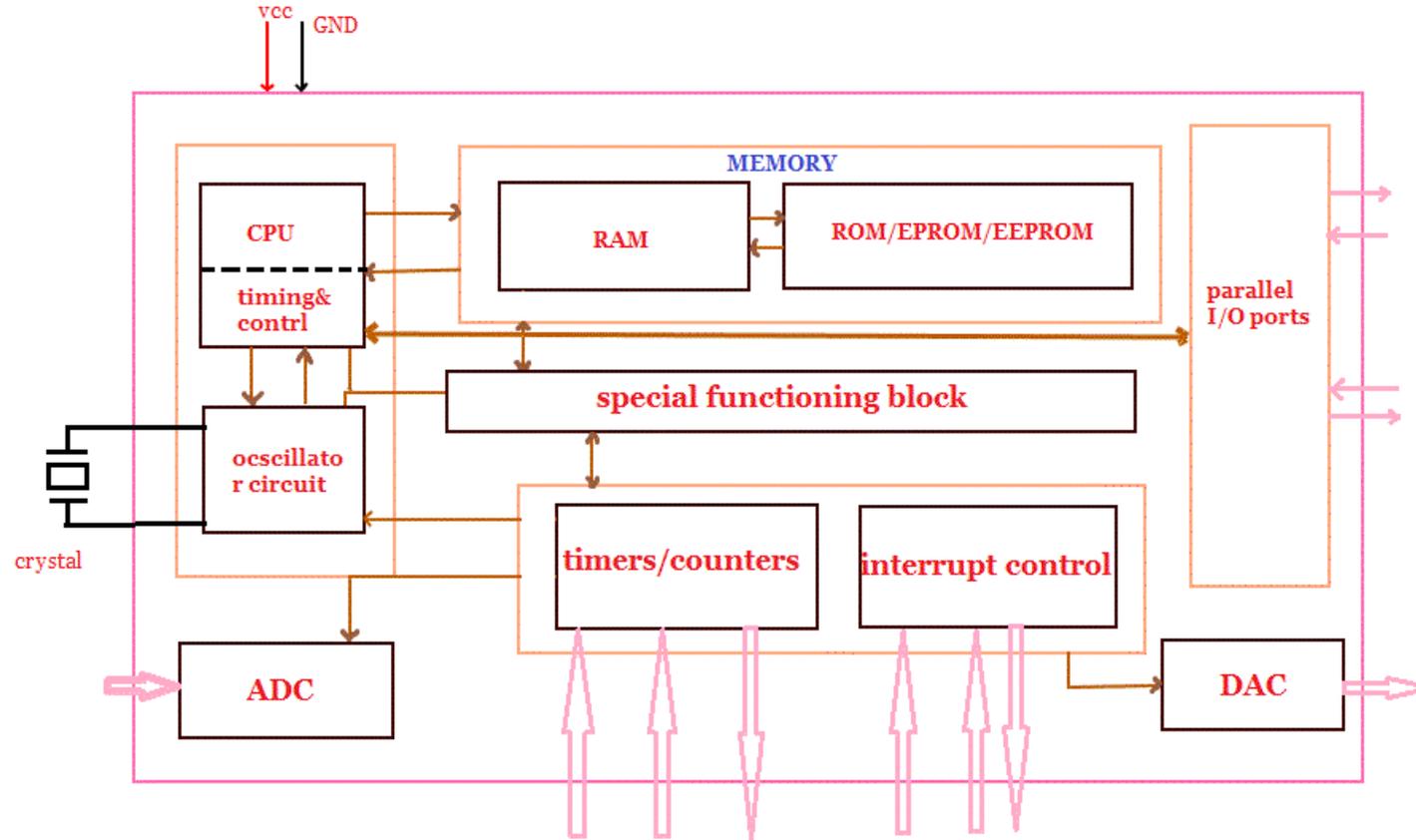
Microcontrollers are used in a wide array of systems and devices. Devices often utilize multiple microcontrollers that work together within the device to handle their respective tasks.

For example, a car might have many microcontrollers that control various individual systems within, such as the anti-lock braking system, traction control, fuel injection or suspension control. All the microcontrollers communicate with each other to inform the correct actions. Some might communicate with a more complex central computer within the car, and others might only communicate with other microcontrollers. They send and receive data using their I/O peripherals and process that data to perform their designated tasks.

1) A microcontroller basically contains one or more following components:

- Central processing unit(CPU)
- Random Access Memory)(RAM)
- Read Only Memory(ROM)
- Input/output ports
- Timers and Counters
- Interrupt Controls
- Analog to digital converters
- Digital analog converters
- Serial interfacing ports
- Oscillatory circuits

Microcontroller structure



CPU

CPU is the **brain of a microcontroller**. CPU is responsible for **fetching the instruction, decodes it, then finally executed**. CPU connects every part of a microcontroller into a single system. The primary function of CPU is fetching and decoding instructions. The instruction fetched from program memory must be decoded by the CPU.

Memory

The function of memory in a microcontroller is the same as a microprocessor. It is used to **store data and program**. A microcontroller usually has a certain amount of RAM and ROM (EEPROM, EPROM, etc) or flash memories for storing program source codes.

Parallel input/output ports

Parallel input/output ports are mainly used to drive/interface various devices such as LCD'S, LED'S, printers, memories, etc to a microcontroller.

Digital to Analog Converter (DAC)

DAC perform reversal operation of ADC conversion. **DAC converts the digital signal into analog format.** It usually used for controlling analog devices like DC motors, various drives, etc.

Interrupt control

The interrupt control used for **providing interrupt (delay)** for a working program. The interrupt may be external (activated by using interrupt pin) or internal (by using interrupt instruction during programming).

Special functioning block

Some microcontrollers used only for some special applications (e.g. space systems and robotics) these controllers containing additional ports to perform such special operations. This considered as special functioning block.

Serial ports

Serial ports provide various serial interfaces between a microcontroller and other peripherals like parallel ports.

Timers/counters

This is the one of the useful function of a microcontroller. A microcontroller may have more than one timer and counters. The timers and counters provide **all timing and counting functions** inside the microcontroller. The major operations of this section are performed **clock functions, modulations, pulse generations, frequency measuring, making oscillations**, etc. This also can be used for counting external pulses.

Analog to Digital Converter (ADC)

ADC converters are used for converting the **analog signal to digital form**. The input signal in this converter should be in analog form (e.g. sensor output) and the output from this unit is in digital form. The digital output can be used for various digital applications (e.g. measurement devices).

Comparison of Microprocessor and Microcontroller

Microprocessors		Microcontrollers
1	It is only a general purpose computer CPU	It is a microcomputer itself
2	Memory, I/O ports, timers, interrupts are not available inside the chip	All are integrated inside the microcontroller chip
3	This must have many additional digital components to perform its operation	Can function as a microcomputer without any additional components.
4	Systems become bulkier and expensive.	Make the system simple, economic and compact
5	Not capable for handling Boolean functions	Handling Boolean functions
6	Higher accessing time required	Low accessing time
7	Very few pins are programmable	Most of the pins are programmable
8	Very few number of bit handling instructions	Many bit handling instructions
9	Widely Used in modern PC and laptops	widely in small control systems

Advantages of Microcontrollers

The main advantages of microcontrollers are given.

- a) Microcontrollers act as a microcomputer without any digital parts.
- b) As the higher integration inside microcontroller reduces cost and size of the system.
- c) Usage of a microcontroller is simple, easy to troubleshoot and system maintaining.
- d) Most of the pins are programmable by the user for performing different functions.
- e) Easily interface additional RAM, ROM, I/O ports.
- f) Low time required for performing operations.

Disadvantages of Microcontrollers

- a) Microcontrollers have got more complex architecture than that of microprocessors.
- b) Only perform a limited number of executions simultaneously.
- c) Mostly used in micro-equipments.
- d) Cannot interface high power devices directly.

Applications of Microcontrollers

Any device that measures, stores, controls, calculates, or displays information must have a microcontroller chip inside. The largest single use for microcontrollers is in the automobile industry (microcontrollers widely used for controlling engines and power controls in automobiles). also find microcontrollers inside keyboards, mouse, modems, printers, and other peripherals. In test equipment, microcontrollers make it easy to add features such as the ability to store measurements, to create and store user routines, and to display messages and waveforms. Consumer products that use microcontrollers include digital camcorders, optical players, LCD/LED display units, etc. And these are just a few examples.

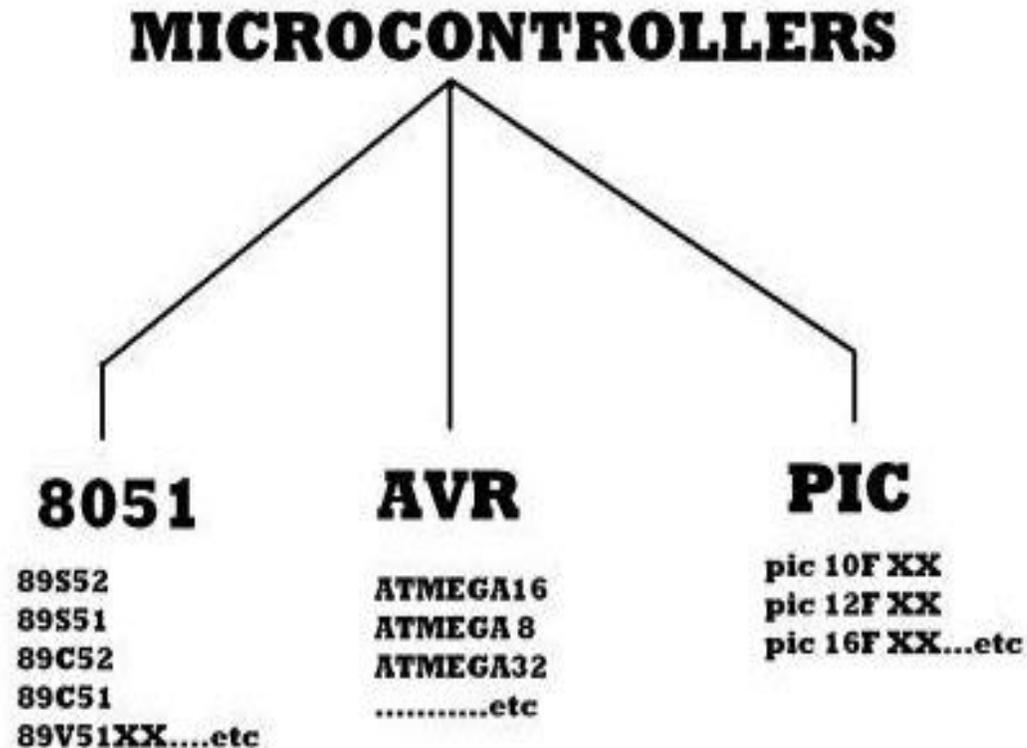
Some basic applications of a microcontroller are given below.

- a) Used in biomedical instruments.
- b) Widely used in communication systems.
- c) Used as a peripheral controller in PC.
- d) Used in robotics.
- e) Used in automobile fields.

How are Microcontrollers Classified?

The microcontrollers are characterized regarding bus-width, instruction set, and memory structure. For the same family, there may be different forms with different sources. This article is going to describe some of the basic types of the Microcontroller that newer users may not know about.

The types of microcontroller is shown in figure, they are characterized by their bits, memory architecture, memory/devices and instruction set. Let's discuss briefly about it.



- ❖ Classification According to Number of Bits
- ❖ Classification According to Memory Devices
- ❖ Classification According to Instruction Set
- ❖ Classification According to Memory Architecture

Classification According to Number of Bits

The bits in microcontroller are 8-bits, 16-bits and 32-bits microcontroller.

In 8-bit microcontroller, the point when the internal bus is 8-bit then the ALU is performs the arithmetic and logic operations. The examples of 8-bit microcontrollers are Intel 8031/8051, PIC1x and Motorola MC68HC11 families.

The 16-bit microcontroller performs greater precision and performance as compared to 8-bit. For example 8 bit microcontrollers can only use 8 bits, resulting in a final range of $0 \times 00 - 0 \times FF$ (0-255) for every cycle. In contrast, 16 bit microcontrollers with its 16 bit data width has a range of $0 \times 0000 - 0 \times FFFF$ (0-65535) for every cycle. A longer timer most extreme worth can likely prove to be useful in certain applications and circuits. It can automatically operate on two 16 bit numbers. Some examples of 16-bit microcontroller are 16-bit MCUs are extended 8051XA, PIC2x, Intel 8096 and Motorola MC68HC12 families.

The 32-bit microcontroller uses the 32-bit instructions to perform the arithmetic and logic operations. These are used in automatically controlled devices including implantable medical devices, engine control systems, office machines, appliances and other types of embedded systems. Some examples are Intel/Atmel 251 family, PIC3x.

Classification According to Memory Devices

The memory devices are divided into two types, they are

1. Embedded memory microcontroller
2. External memory microcontroller

Embedded memory microcontroller: When an embedded system has a microcontroller unit that has all the functional blocks available on a chip is called an embedded microcontroller. For example, 8051 having program & data memory, I/O ports, serial communication, counters and timers and interrupts on the chip is an embedded microcontroller.

External Memory Microcontroller: When an embedded system has a microcontroller unit that has not all the functional blocks available on a chip is called an external memory microcontroller. For example, 8031 has no program memory on the chip is an external memory microcontroller

Classification According to Instruction Set

CISC: CISC is a Complex Instruction Set Computer. It allows the programmer to use one instruction in place of many simpler instructions.

RISC: The RISC is stands for Reduced Instruction set Computer, this type of instruction sets reduces the design of microprocessor for industry standards. It allows each instruction to operate on any register or use any addressing mode and simultaneous access of program and data.

Example for CISC and RISC:

CISC:	Mov AX, 4	RISC:		Mov AX, 0
	Mov BX, 2			Mov BX, 4
	ADD BX, AX			Mov CX, 2
			Begin	ADD AX, BX
			Loop	Begin

From above example, RISC systems shorten execution time by reducing the clock cycles per instruction and CISC systems shorten execution time by reducing the number of instructions per program. The RISC gives a better execution than the CISC.

Classification According to Memory Architecture

Memory architecture of microcontroller are two types, they are namely:

1. Harvard memory architecture microcontroller
2. Princeton memory architecture microcontroller

Harvard Memory Architecture Microcontroller: The point when a microcontroller unit has a dissimilar memory address space for the program and data memory, the microcontroller has Harvard memory architecture in the processor.

Princeton Memory Architecture Microcontroller: The point when a microcontroller has a common memory address for the program memory and data memory, the microcontroller has Princeton memory architecture in the processor.

Memory

The function of memory in a microcontroller is the same as a microprocessor. It is used to store data and program. A microcontroller usually has a certain amount of RAM and ROM (EEPROM, EPROM, etc) or flash memories for storing program source codes.

Therefore, an obvious distinction of memory types can be made according to their function :

Register File: A (usually) relatively small memory embedded on the CPU. It is used as a scratchpad for temporary storage of values the CPU is working with - you could call it the CPU's short term memory.

Data Memory: For longer term storage, generic CPUs usually employ an external memory which is much larger than the register file. Data that is stored there may be short-lived, but may also be valid for as long as the CPU is running. Of course, attaching external memory to a CPU requires some hardware effort and thus incurs some cost. For that reason, microcontrollers usually sport on-chip data memory.

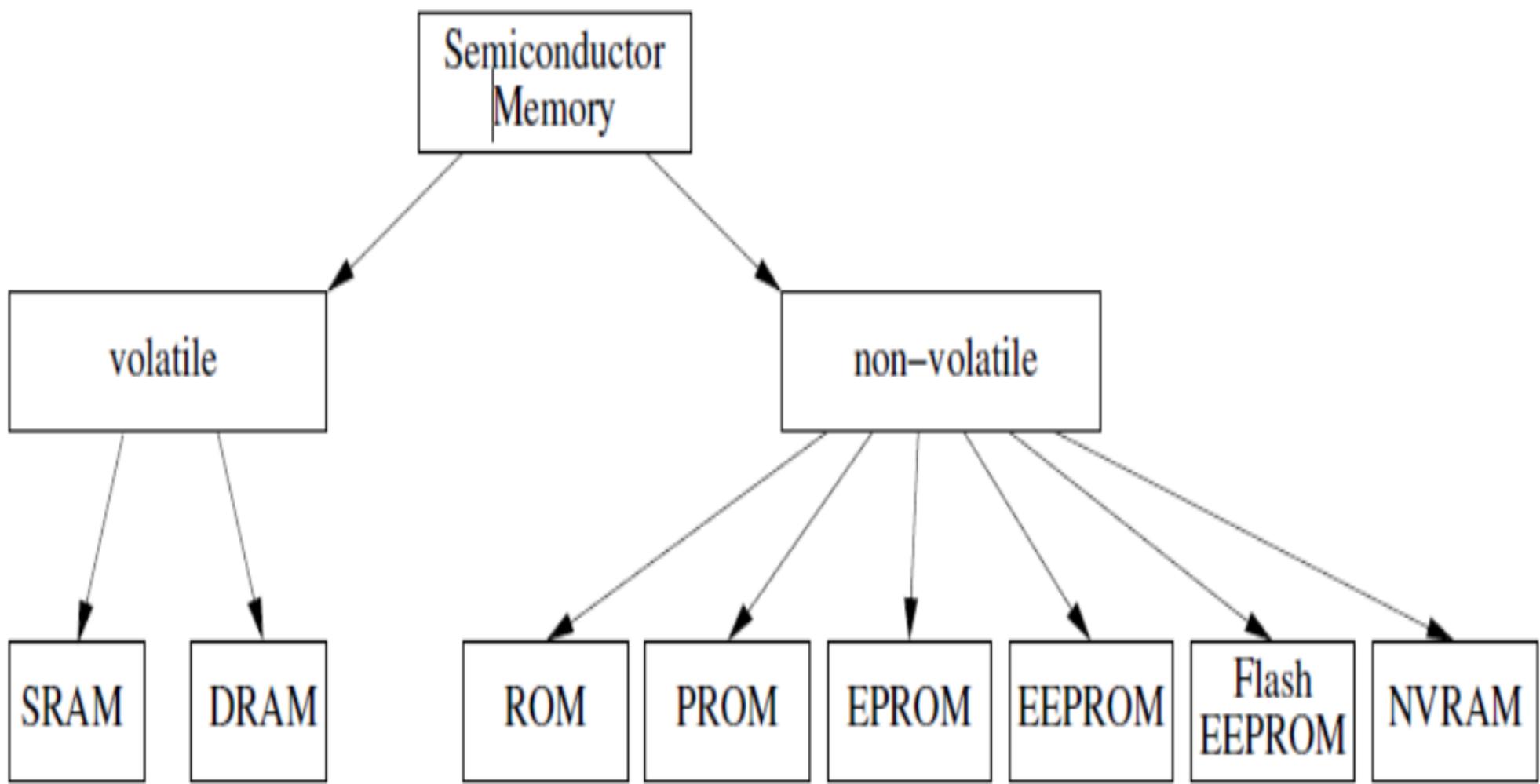
Instruction Memory: Like the data memory, the instruction memory is usually a relatively large external memory (at least with general CPUs). Actually, with von-Neumann-architectures, it may even be the same physical memory as the data memory. With microcontrollers, the instruction memory, too, is usually integrated right into the MCU.

There, the most basic distinction would be **volatile versus non-volatile memory**. In this context, volatile means that the contents of the memory are lost as soon as the system's power is switched off.

Of course, there are different ways either type of memory can be implemented. Therefore, the distinction based on the physical properties can go into more detail. Volatile memory can be static or dynamic, and there is quite a variety of non-volatile memory types: ROM, PROM, EPROM, EEPROM, FLASH, NV-RAM. Let's examine those more closely.

Non-volatile Memory

Contrary to SRAMs and DRAMs, non-volatile memories retain their content even when power is cut. But, as already mentioned, that advantage comes at a price: Writing non-volatile memory types is usually much slower and comparatively complicated, often downright annoying.



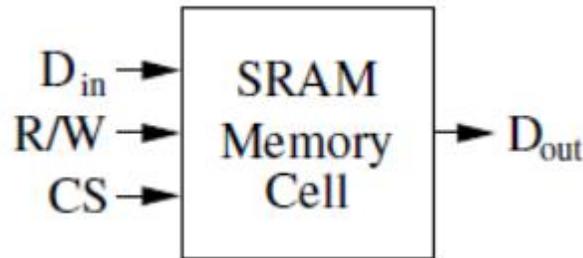
Volatile Memory

As mentioned above, volatile memory retains its contents only so long as the system is powered on. Then why should you use volatile memory at all, when non-volatile memory is readily available?

The problem here is that non-volatile memory is usually a lot slower, more involved to work with, and much more expensive. While the volatile memory in your PC has access times in the nanosecond range, some types of non-volatile memory will be unavailable for milliseconds after writing one lousy byte to them.

Static RAM

Disregarding the era of computers before the use of integrated circuits, Static Random Access Memory (SRAM) was the first type of volatile memory to be widely used. An SRAM chip consists of an array of cells, each capable of storing one bit of information. To store a bit of information, a so-called flip-flop is used, which basically consists of six transistors. For now, the internal structure of such a cell is beyond the scope of our course, so let's just view the cell as a black box.



Dynamic RAM

In contrast to a well known claim that nobody will ever need more than 640 kilobytes of RAM, there never seems to be enough memory available. Obviously, we would like to get as much storage capacity as possible out of a memory chip of a certain size.

That is what was achieved with Dynamic Random Access Memory – DRAM: The number of transistors needed per bit of information was brought down to one. This, of course, reduced the silicon area for a given cell count. So at the same chip size, a DRAM has much larger storage capacity compared to an SRAM.

Non-volatile Memory

Contrary to SRAMs and DRAMs, non-volatile memories retain their content even when power is cut. But, as already mentioned, that advantage comes at a price: Writing non-volatile memory types is usually much slower and comparatively complicated, often downright annoying.

1.ROM

2.PROM

3.EPROM

4.EEPROM

6.Flash

7.NVRAM

ROM

Read Only Memories (ROMs) were the first types of non-volatile semiconductor memories. Did we just say write access is more involved with non-volatile than with volatile memory? Well, in the case of ROM, we kind of lied: As the name implies, you simply cannot write to a ROM. If you want to use ROMs, you have to hand the data over to the chip manufacturer, where a specific chip is made containing your data.

Programmable Read Only Memory (PROM)

These are basically matrices of memory cells, each containing a silicon fuse. Initially, each fuse is intact and each cell reads as a logical 1. By selecting a cell and applying a short but high current pulse, the cell's fuse can be destroyed, thereby programming a logical 0 into the selected cell.

Erasable Programmable Read Only Memory (EPROM)

Here, programming is non-destructive. Memory is stored in so-called field effect transistors (FETs), or rather in one of their pins called gate. It is aptly named floating gate, as it is completely insulated from the rest of the circuit. However, by applying an appropriately high voltage, it is possible to charge the floating gate via a physical process called avalanche injection. So, instead of burning fuses, electrons are injected into the floating gate, thus closing the transistor switch.

EEPROM (Electrically Erasable and Programmable ROM)

has all the advantages of an EPROM without the hassle. No special voltage is required for programming anymore, and – as the name implies – no more UV light source is needed for erasing. EEPROM works very similar to EPROM, except that the electrons can be removed from the floating gate by applying an elevated voltage.

Flash

is a variant of EEPROM where erasing is not possible for each address, but only for larger blocks or even the entire memory (erased ‘in a flash’, so to speak). That way, the internal logic is simplified, which in turn reduces the price considerably. Also, due to the fact that it is not possible to erase single bytes, Flash EEPROM is commonly used for program, not data memory.

Non-Volatile RAM (NVRAM).

This can be achieved in different ways. One is to just add a small internal battery to an SRAM device, so that when external power is switched off, the SRAM still retains its content. Another variant is to combine a SRAM with an EEPROM in one package. Upon power-up, data is copied from the EEPROM to the SRAM. During operation, data is read from and written to the SRAM. When power is cut off, the data is copied to the EEPROM.

Automation:

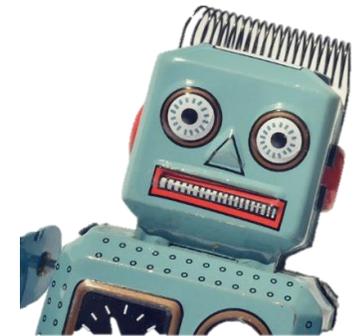
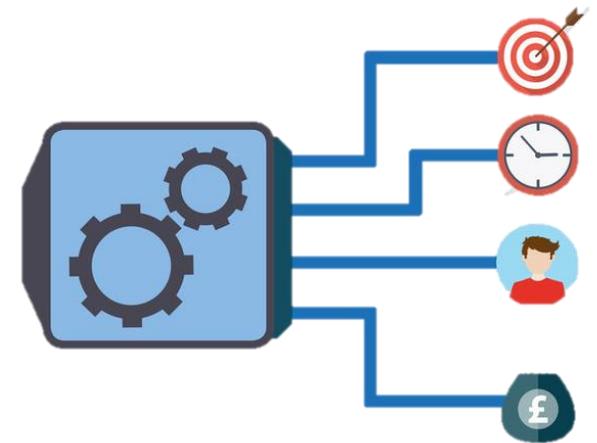
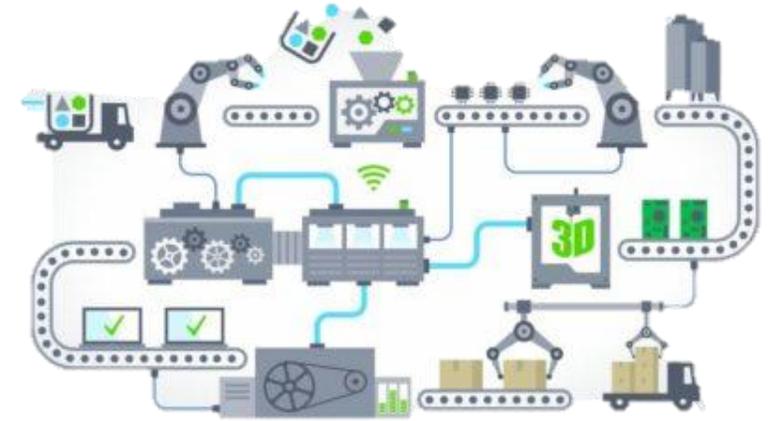
Automation system components, Industrial communications, Continuous control.

Introduction of PLC, PLC Hardware, Definitions of Allen-Bradley conditional inputs and outputs, I/O configuration, Programming Terminals and Peripherals, Application of PLC in Automation,

Microcontroller basics, classification, basic Architecture, memory, registers, I/O ports.

overview of SCADA and DCS systems;

AUTOMATION OVERVIEW



AUTOMATION SYSTEM

Programmable Logic Controller (PLC)



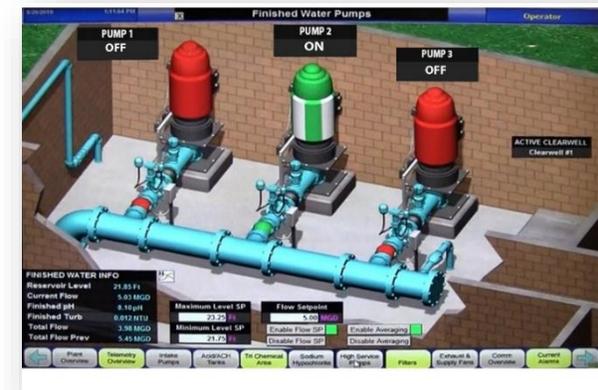
Human Machine Interface (HMI)



Variable-Frequency Drive (VFD) / DC Drive



Supervisory Control and Data Acquisition (SCADA)



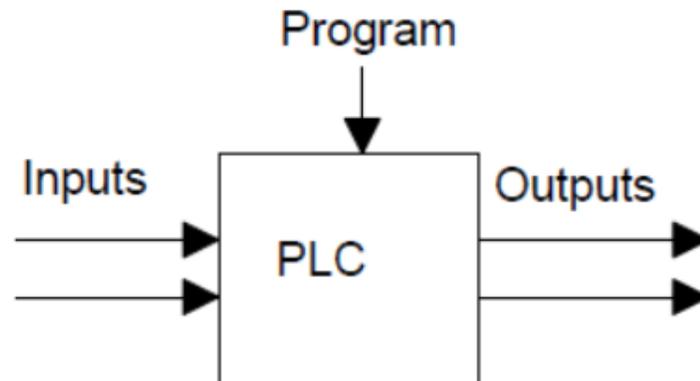
Instrumentation:



programmable logic controller (PLC)

A programmable logic controller (PLC) is a special form of microprocessor- based controller that uses a programmable memory to store instructions and to implement functions such as logic, sequencing, timing, counting and arithmetic in order to control machines and processes and are designed to be operated by engineers with perhaps a limited knowledge of computers and computing languages.

The first PLC was developed in 1969. They are now widely used and extend from small self-contained units for use with perhaps 20 digital inputs/outputs to modular systems which can be used for large numbers of inputs/outputs, handle digital or analogue inputs/outputs, and also carry out proportional-integral-derivative control modes.



AUTOMATION SYSTEM

Programmable Logic Controller (PLC):



Typically a PLC system has the basic functional components of processor unit, memory, power supply unit, input/output interface section, communications interface and the programming device.

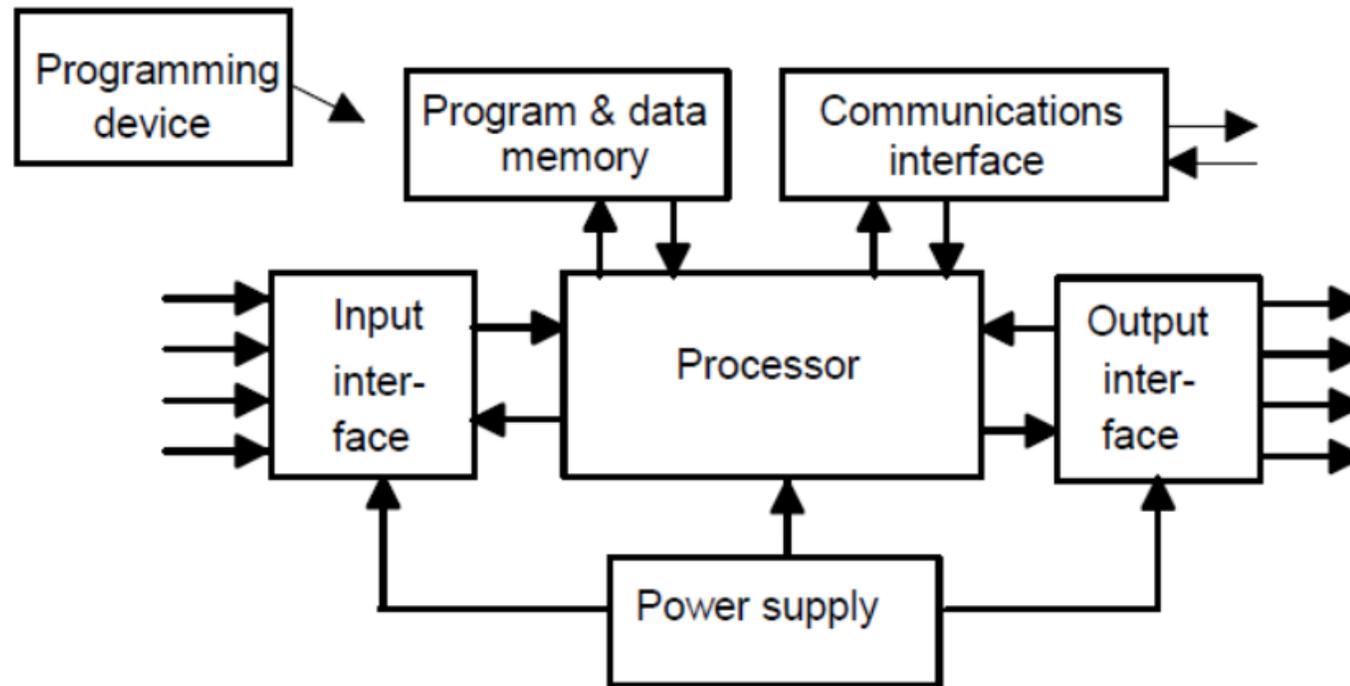


Fig : The PLC system

1. The processor unit or central processing unit (CPU) is the unit containing the microprocessor and this interprets the input signals and carries out the control actions, according to the program stored in its memory, communicating the decisions as action signals to the outputs.
2. The power supply unit is needed to convert the mains a.c. voltage to the low d.c. voltage (5 V) necessary for the processor and the circuits in the input and output interface modules.
3. The programming device is used to enter the required program into the memory of the processor. The program is developed in the device and then transferred to the memory unit of the PLC.

4. The **memory unit is where the program is stored** that is to be **used for the control actions** to be exercised by the microprocessor and data stored from the input for processing and for the output for outputting.

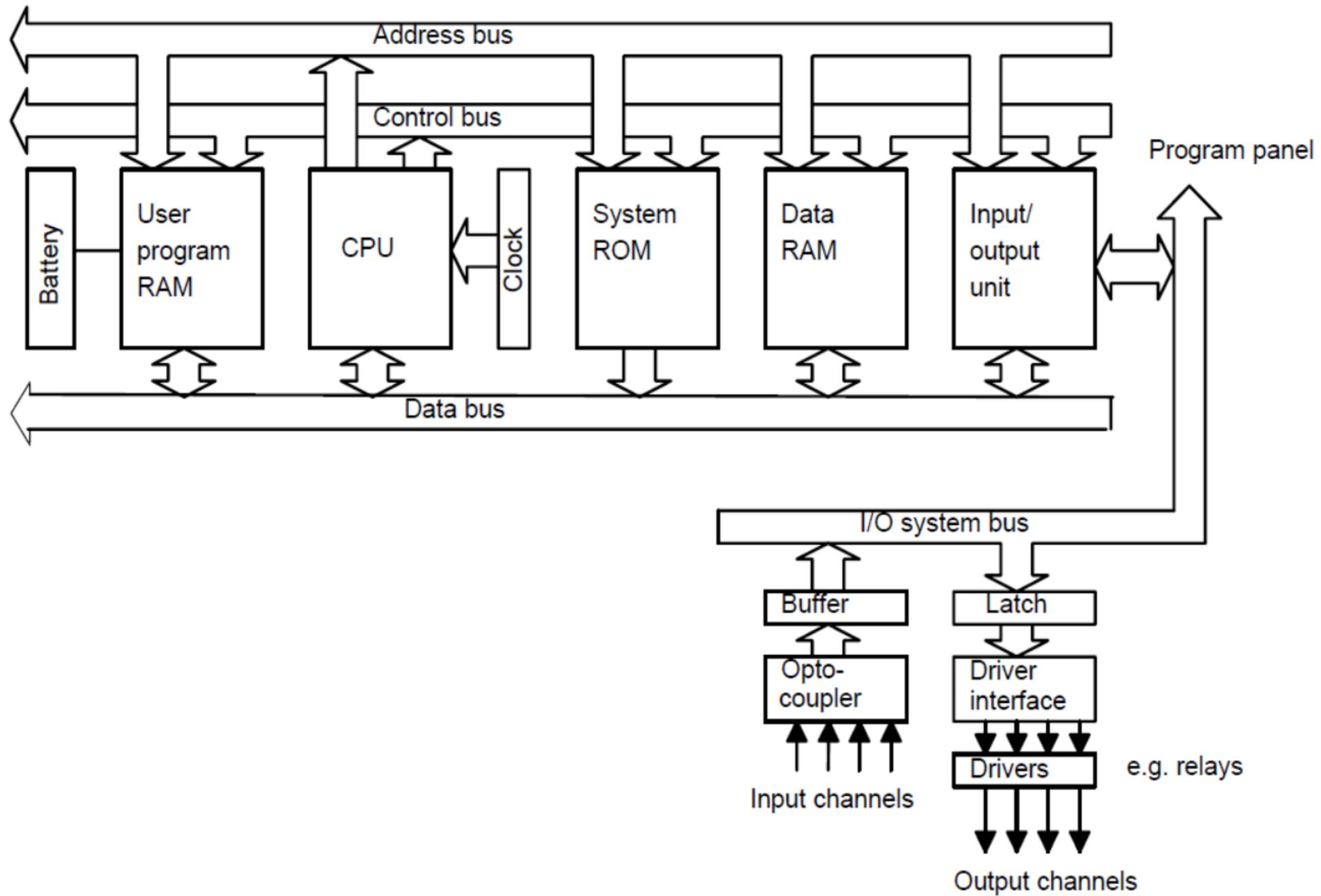
5. The input and output sections are where the processor **receives information from external devices** and **communicates information to external** devices.

6. The communications interface is **used to receive and transmit data on communication networks from or to other remote PLCs** .It is concerned with such actions as device verification, data acquisition, synchronization between user applications and connection management.

Internal architecture of 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 synchronization 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. The term word is used for the group of bits constituting some information. Thus an 8-bit word might be the binary number 00100110. 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**.
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 synchronize actions.
- 4 The **system bus is used for communications between the input/output ports and the input/output unit**.

Memory

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..**
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 optoisolators (the term *optocoupler* is also often used).

Sourcing and sinking

The terms sourcing and sinking are used to describe the way in which d.c. devices are connected to a PLC. With sourcing, using the conventional current flow direction as from positive to negative, an input device receives current from the input module, i.e. the input module is the source of the current. If the current flows from the output module to an output load then the output module is referred to as sourcing. With sinking, using the conventional current flow direction as from positive to negative, an input device supplies current to the input module,

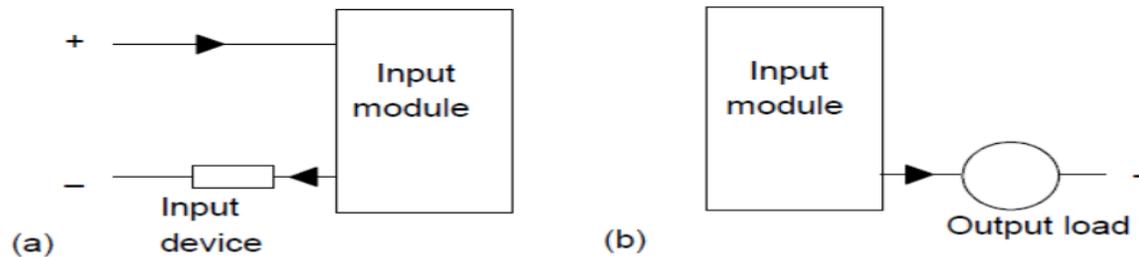


Figure 1.11 *Sourcing*

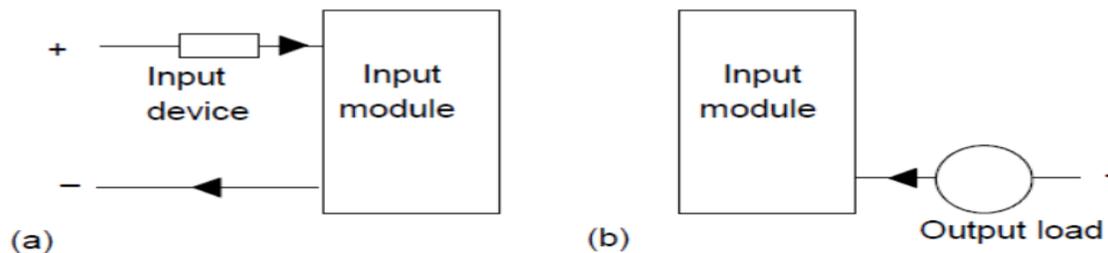


Figure 1.12 *Sinking*

PLC systems

There are two common types of mechanical design for PLC systems;

1. A single box and
2. The modular/rack types.

1. The single box type (or, as sometimes termed, brick) is commonly **used for small programmable controllers** and is supplied as an integral compact package complete with power supply, processor, memory, and input/output units. Typically such a PLC might have 6, 8, 12 or 24 inputs and 4, 8 or 16 outputs and a memory which can store some 300 to 1000 instructions.

2. The modular type consists of **separate modules for power supply, processor, etc.**, which are often mounted on rails within a metal cabinet. The rack type **can be used for all sizes of programmable controllers and has the various functional units packaged in individual modules** which can be plugged into sockets in a base rack. The mix of modules required for a particular purpose is decided by the user and the appropriate ones then plugged into the rack. Thus it is comparatively easy to expand the number of input/output (I/O) connections by just adding more input/output modules or to expand the memory by adding more memory units.

An example of such a modular system is provided by the Allen-Bradley PLC-5 PLC of Rockwell automation (Figure 1.15). PLC-5 processors are available in a range of I/O capacity and memory size, and can be configured for a variety of communication networks.

Programming PLC

Programming devices can be a hand-held device, a desktop console or a computer. Only when the program has been designed on the programming device and is ready is it transferred to the memory unit of the PLC.

1. Hand-held programming devices will normally contain enough memory to allow the unit to retain programs while being carried from one place to another.
2. Desktop consoles are likely to have a visual display unit with a full keyboard and screen display.
3. Personal computers are widely configured as program development workstations. Some PLCs only require the computer to have appropriate software; others require special communication cards to interface with the PLC. A major advantage of using a computer is that the program can be stored on the hard disk or a CD and copies easily made.

Input–output devices

The input devices considered include digital and analogue devices such as mechanical switches for position detection, proximity switches, photoelectric switches, encoders, temperature and pressure switches, potentiometers, linear variable differential transformers, strain gauges, thermistors, thermo transistors and thermocouples.

Output devices considered include relays, contactors, solenoid valves and motors.

Input devices

The term sensor is used for an input device that provides a usable output in **response to a specified physical input**.

For example, a **thermocouple** is a sensor which converts a temperature difference into an electrical output. The term transducer is generally used for a device that converts a signal from one form to a **different physical form**. Thus sensors are often transducers, but also other devices can be transducers, e.g. a motor which converts an electrical input into rotation.

Sensors which give digital/discrete, i.e. on–off, outputs can be easily connected to the input ports of PLCs. Sensors which give analogue signals have to be converted to digital signals before inputting them to PLC ports.

Common terms used to define the performance of sensors.

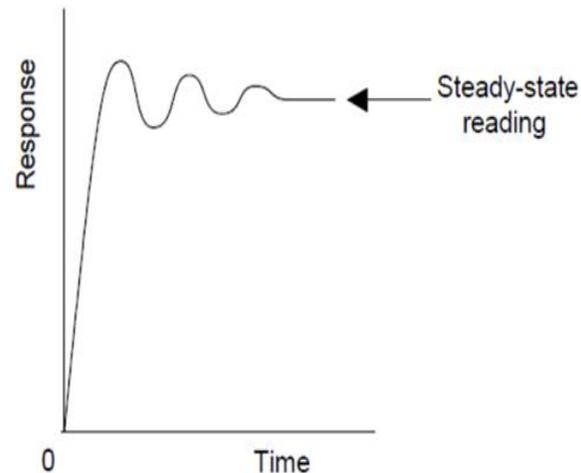
1. Accuracy is the extent(limit) to which the value indicated by a measurement system or element might be wrong. For example, a temperature sensor might have an accuracy of $\pm 0.1^{\circ}\text{C}$.
2. The range of variable of system is the limits between which the input can vary. For example, a resistance temperature sensor might be quoted as having a range of -200 to $+800^{\circ}\text{C}$.

3. When the input value to a sensor changes, it will take some time to reach and settle down to the steady-state value .

The **response time** is the time which elapses(pass) after the input to a system or element is abruptly increased from zero to a constant value up to the point at which the system or element gives an output corresponding to some specified percentage, e.g. 95%, of the value of the input .

The **rise time** is the time taken for the output to rise to some specified percentage of the steady-state output. Often the rise time refers to the time taken for the output to rise from 10% of the steady-state value to 90 or 95% of the steady-state value.

The **settling time** is the time taken for the output to settle to within some percentage, e.g. 2%, of the steady-state value.



4 .The **sensitivity** indicates **how much the output** of an instrument system or system element **changes** when the quantity being measured changes **by** a given amount, i.e. the ratio output/input. For example, a thermocouple might have a sensitivity of $20 \mu\text{V}/\text{oC}$ and so give an output of $20 \mu\text{V}$ for each 1°C change in temperature.

5. The **stability** of a system is its **ability to give the same output** when used to measure a **constant input** over a period of time. The term **drift** is often used to describe the **change in output that occurs over time**. The drift may be expressed as a percentage of the full range output. The term zero drift is used for the changes that occur in output when there is zero input.

6 .The **term repeatability** is used for the ability of a measurement system to give the same value **for repeated measurements of the same value of a variable**. Common cause of lack of repeatability are random fluctuations in the environment, e.g. changes in temperature and humidity. The error arising from repeatability is usually expressed as a percentage of the full range output. For example, a pressure sensor might be quoted as having a repeatability of $\pm 0.1\%$ of full range. Thus with a range of 20 kPa this would be an error of ± 20 Pa.

7. The **reliability** of a measurement system, or element in such a system, is defined as **being the probability** that it will operate to an agreed level of performance, for a specified period, subject to specified environmental conditions. The agreed level of performance might be that the measurement system gives a particular accuracy.

commonly used PLC input devices and their sensors.

1. Mechanical switches
2. Proximity switches
3. Photoelectric sensors and switches
4. Encoders
5. Temperature sensors
6. Position/displacement sensors
7. Strain gauges
8. Pressure sensors
9. Liquid level detector
10. Fluid flow measurement
11. Smart sensors

Output devices

The output ports of a PLC are of the relay type or optoisolator with transistor or triac types depending on the devices connected to them which are to be switched on or off. Generally, the digital signal from an output channel of a PLC is used to control an actuator which in turn controls some process. The term actuator is used for the device which transforms the electrical signal into some more powerful action which then results in the control of the process. The following are some examples.

1. Relay
2. Directional control valves
3. Motors
4. Stepper motors

Motors Control by plc

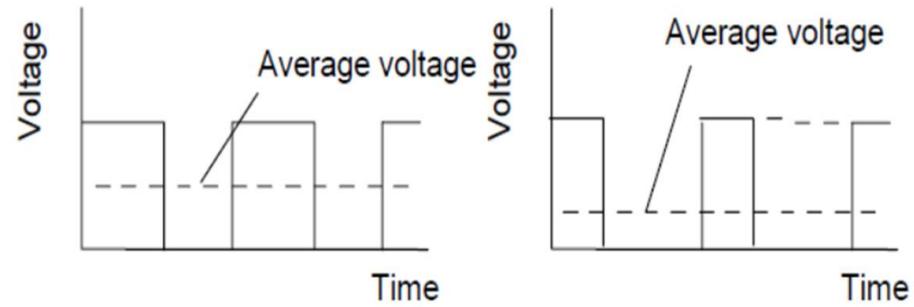


Figure : Pulse width modulation

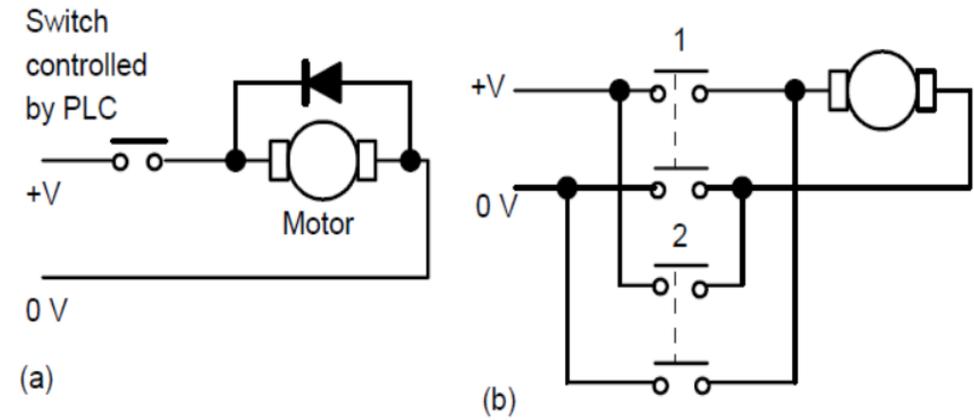
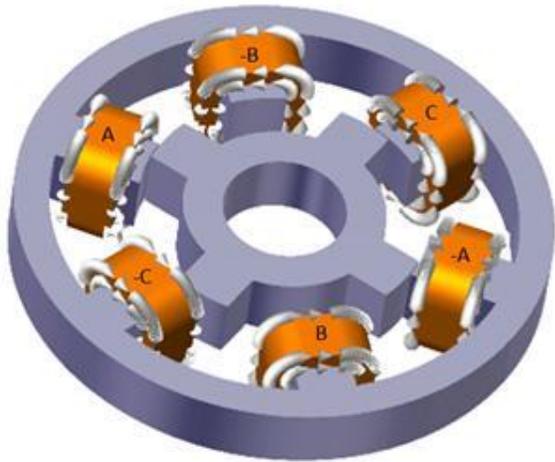


Figure D.c. motor: (a) on-off control, (b) directional control

Stepper motors

A stepper motor is a type of DC motor that works in discrete steps. It is a synchronous brushless motor where a full rotation is divided into a number of steps. ... The rotor is the rotating shaft and the stator consists of electromagnets that form the stationary part of the motor.



Stepper Motor Applications:

As the stepper motor are digitally controlled using an input pulse, they are suitable for use with computer controlled systems. They are used in numeric control of machine tools. Used in tape drives, floppy disc drives, printers and electric watches.

Advantages of Stepper Motor:

1. The rotation angle of the motor is proportional to the input pulse.
2. The motor has full torque at standstill.
3. Precise positioning and repeatability of movement since good stepper motors have an accuracy of 3 – 5% of a step and this error is non cumulative from one step to the next.
4. Excellent response to starting, stopping and reversing.
5. Very reliable since there are no contact brushes in the motor. Therefore the life of the motor is simply dependent on the life of the bearing.
6. The motors response to digital input pulses provides open-loop control, making the motor simpler and less costly to control.
7. It is possible to achieve very low speed synchronous rotation with a load that is directly coupled to the shaft.
8. A wide range of rotational speeds can be realized as the speed is proportional to the frequency of the input pulses

Stepper motors

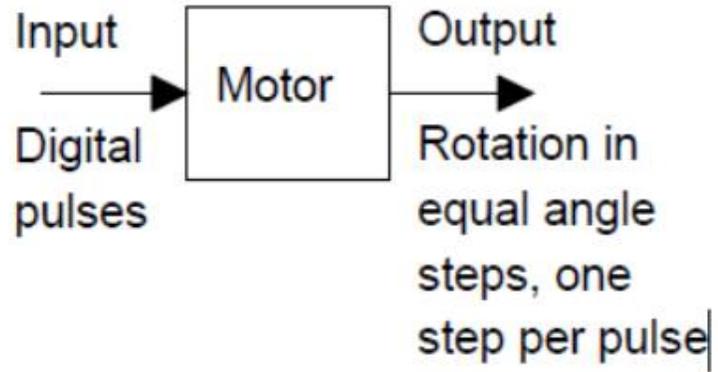


Figure .The stepping motor

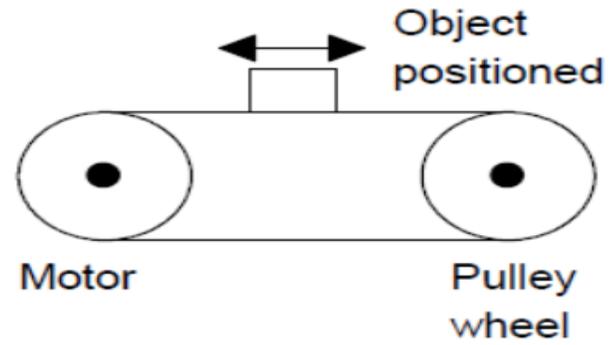


Figure Linear positioning

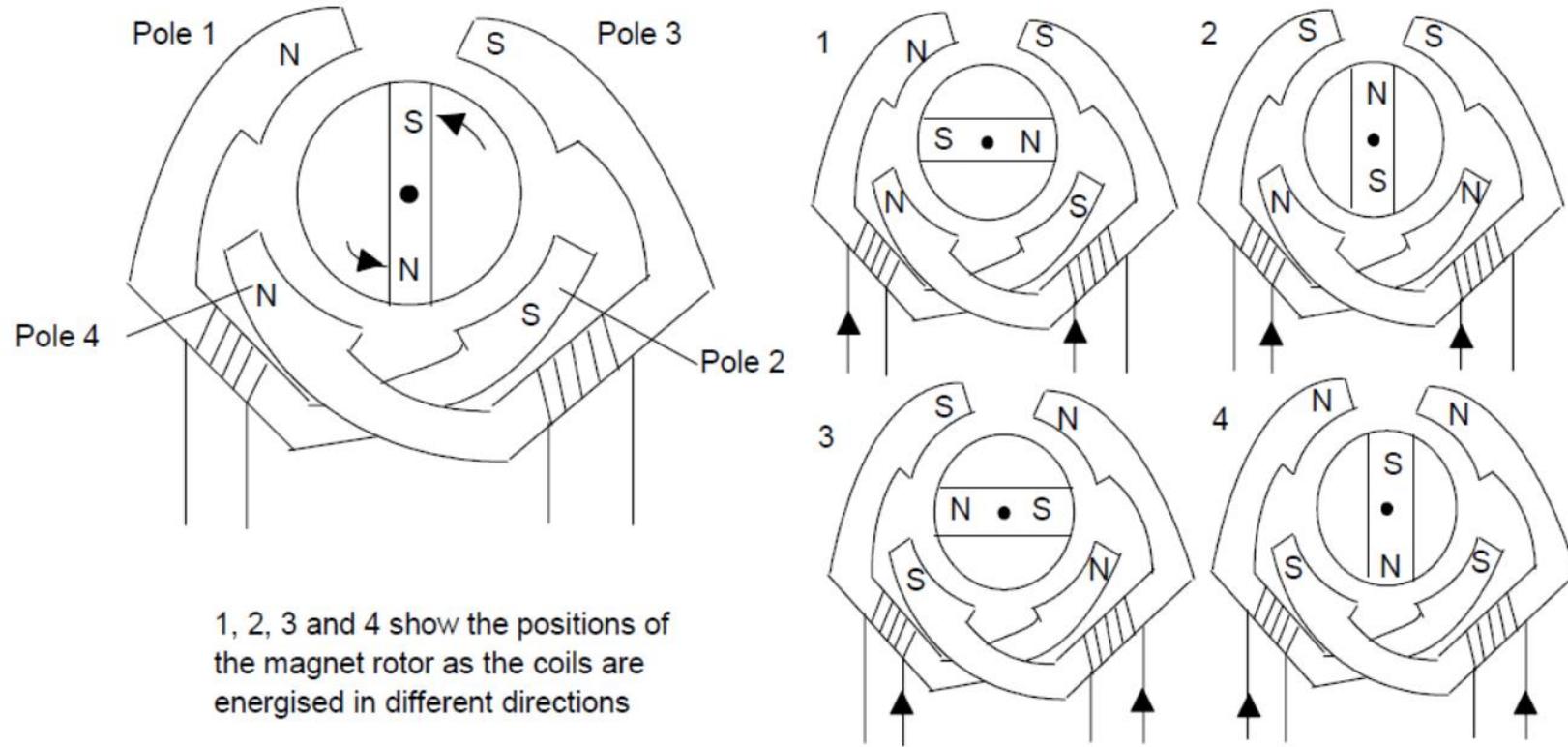


Figure :The basic principles of the permanent magnet stepper motor (2-phase) with 90 o steps

Step	Pole 1	Pole 2	Pole 3	Pole 4
1	North	South	South	North
2	South	North	South	North
3	South	North	North	South
4	North	South	North	South
5	Repeat of steps 1 to 4			

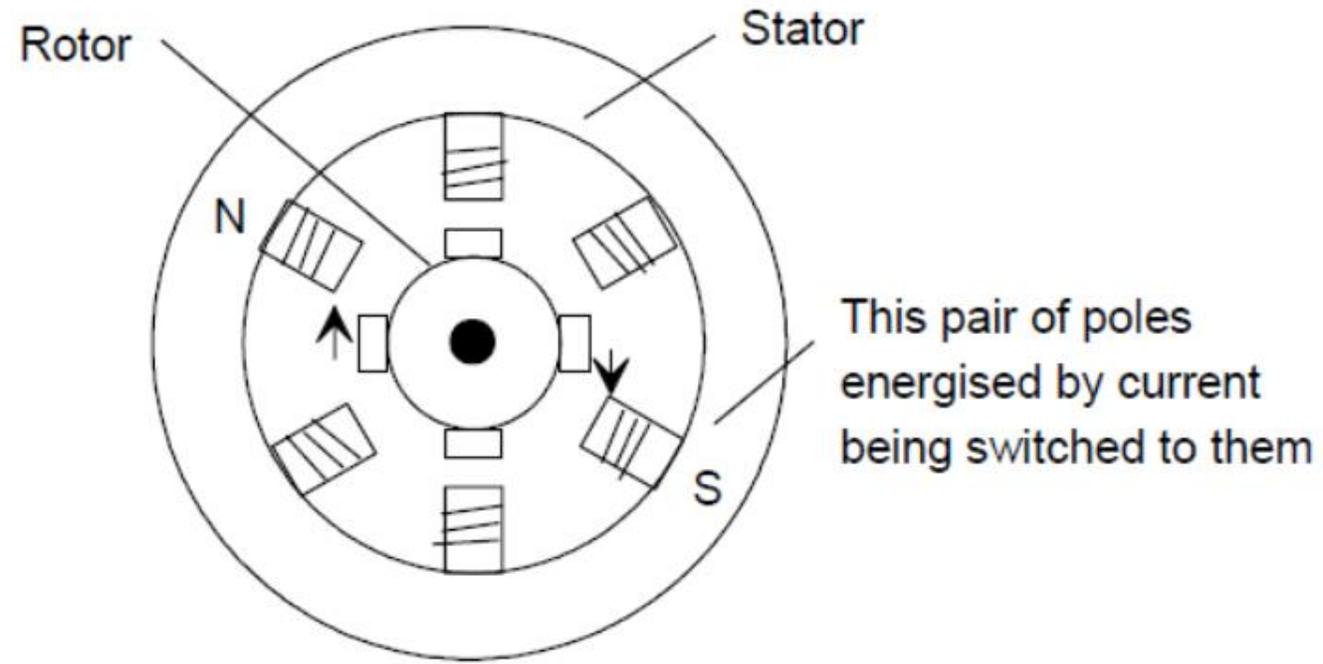


Figure : The principle of the variable reluctance stepper motor

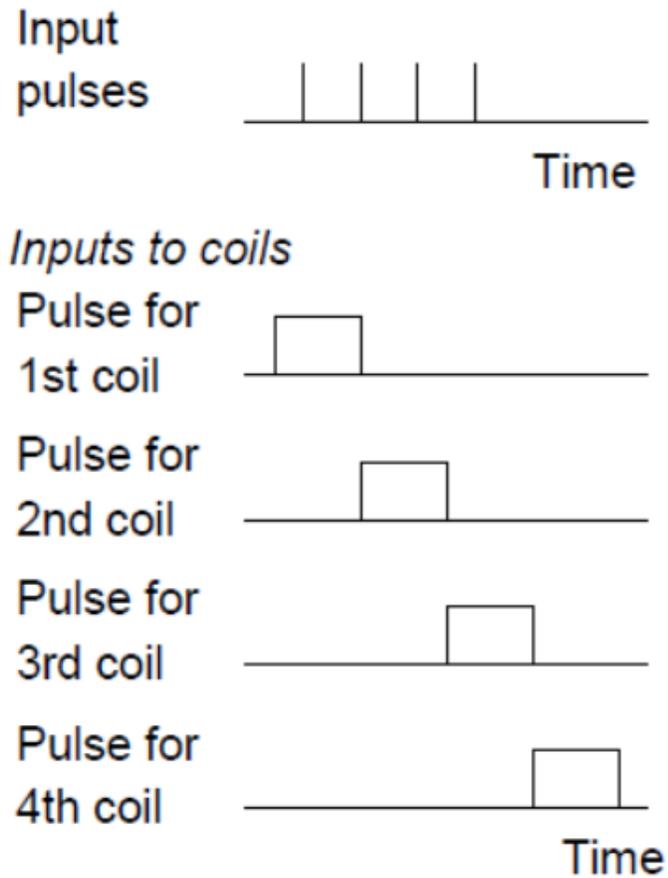


Figure : Input and outputs of the drive system

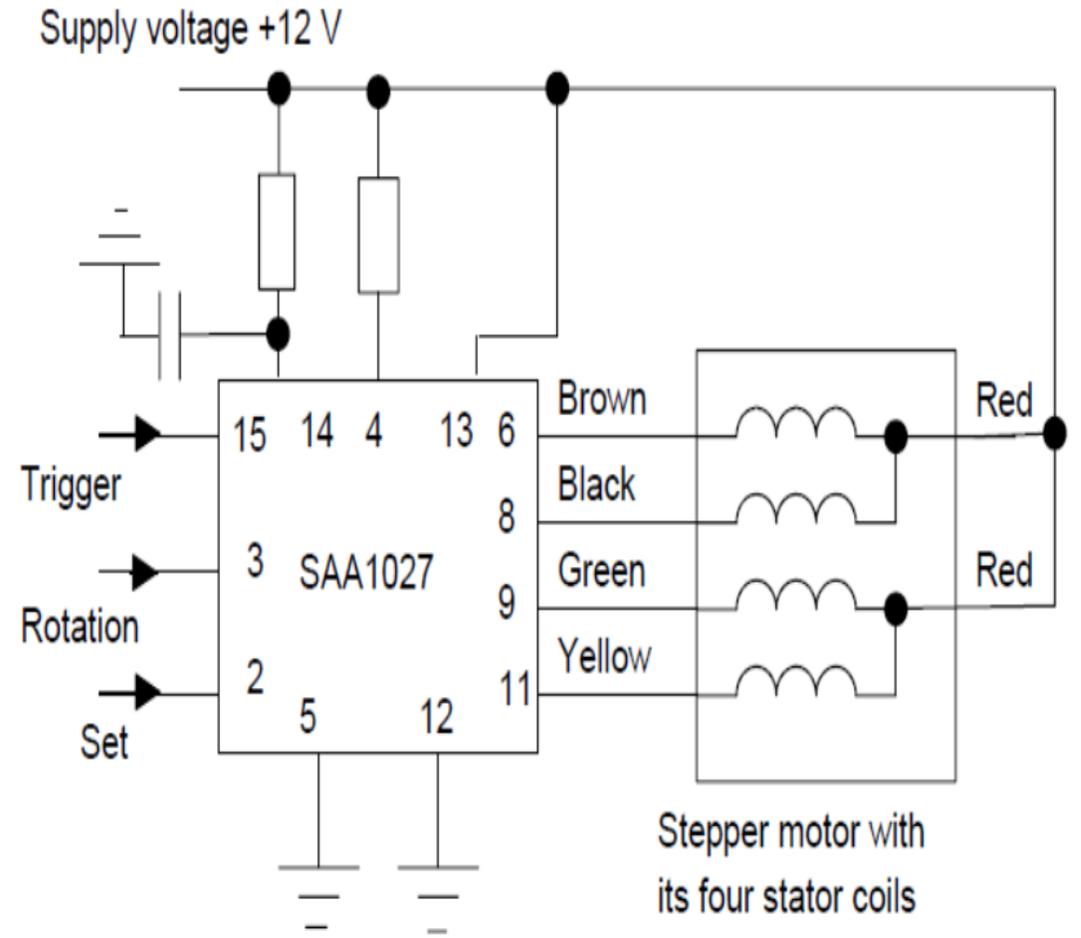
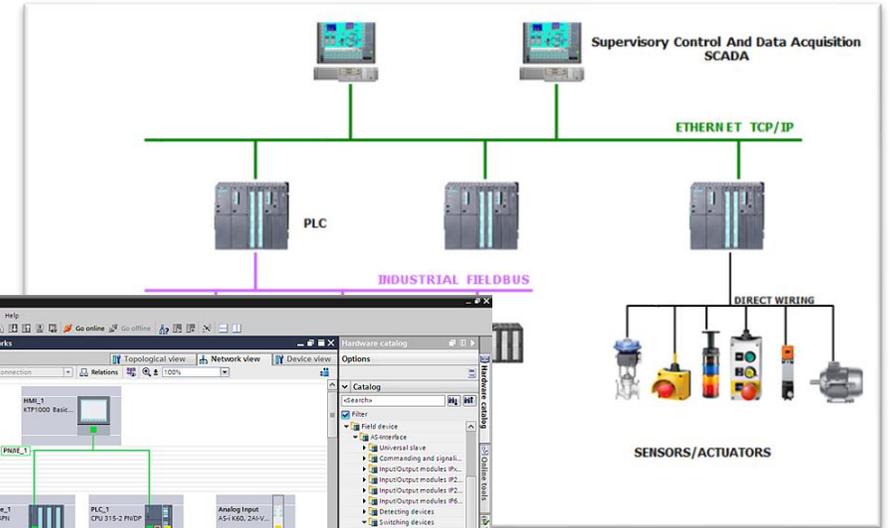


Figure :Driver circuit connections with the integrated circuit SAA1027

AUTOMATION SYSTEM

Programmable Logic Controller (PLC):



The screenshot displays the Siemens SIMATIC Manager software interface. The top window shows the AS-Interface configuration, including a network diagram with a PLC (CPU 315-2 DP) and various modules like the ID device, Compact Starter, and Analog Input. Below this, several ladder logic programs are visible, including:

- MainProgram - InsertionSort1:** A program implementing an insertion sort algorithm. The code includes comments and logic for sorting an array of numbers.
- MainProgram - SD7_MachineConditions1:** A program for machine conditions, featuring a state transition diagram with states like 'Remote_Mode' and 'Made_Toggle_PD'.
- MainProgram - Rcvp_Sweet_Cream.Op.2:** A program for a specific operation, showing a state transition diagram with states like 'Tran_000' and 'Tran_001'.

The interface also shows a project tree on the left, a details view for the selected program, and a hardware catalog on the right.

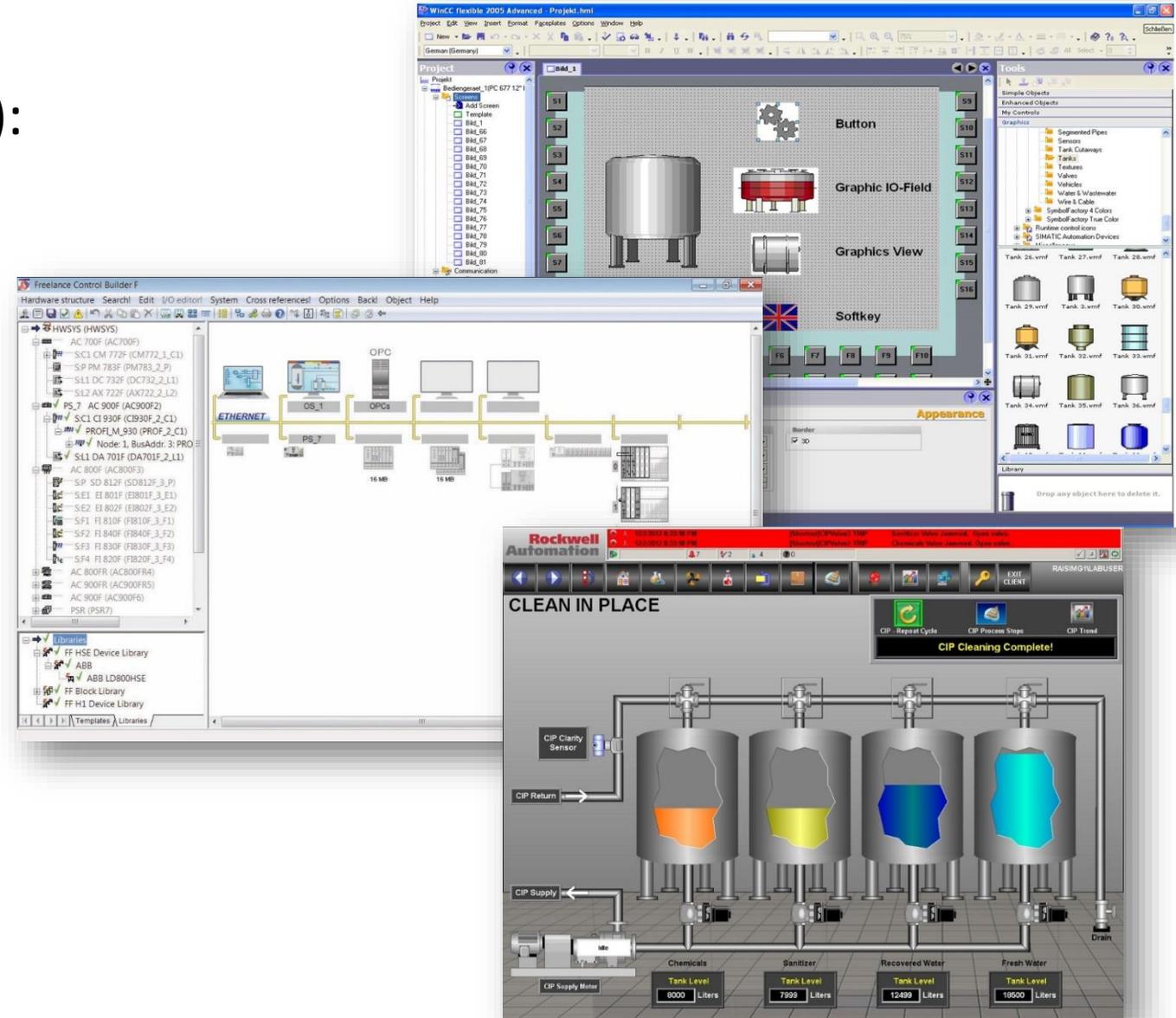
AUTOMATION SYSTEM

Human Machine Interface (HMI):



AUTOMATION SYSTEM

Human Machine Interface (HMI):



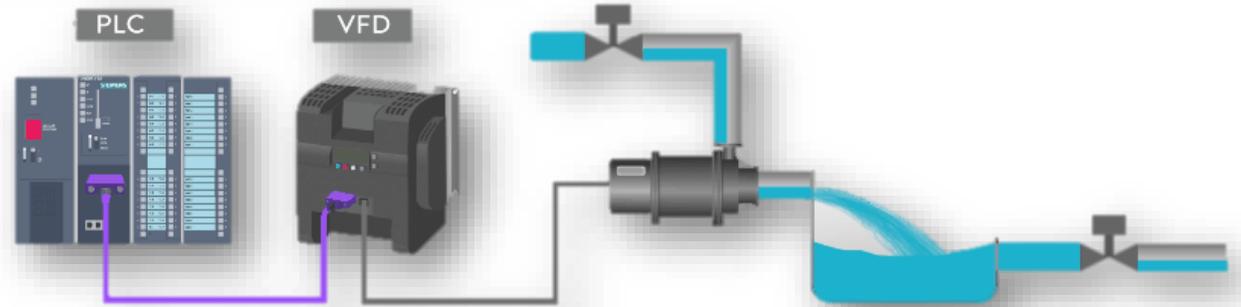
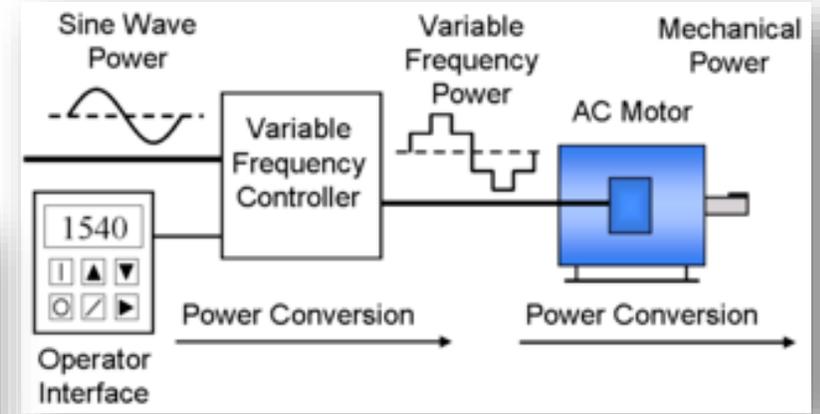
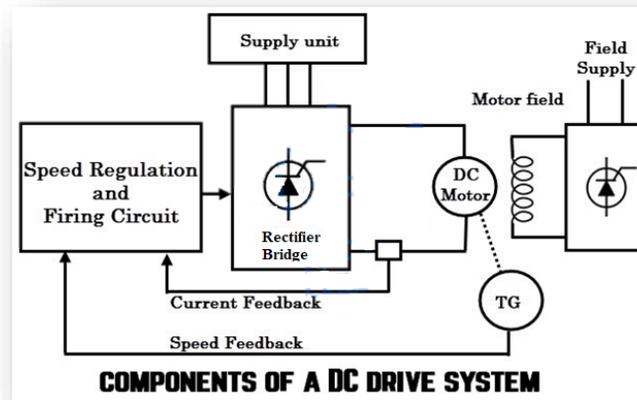
AUTOMATION SYSTEM

Variable-Frequency Drive (VFD) / DC Drive:



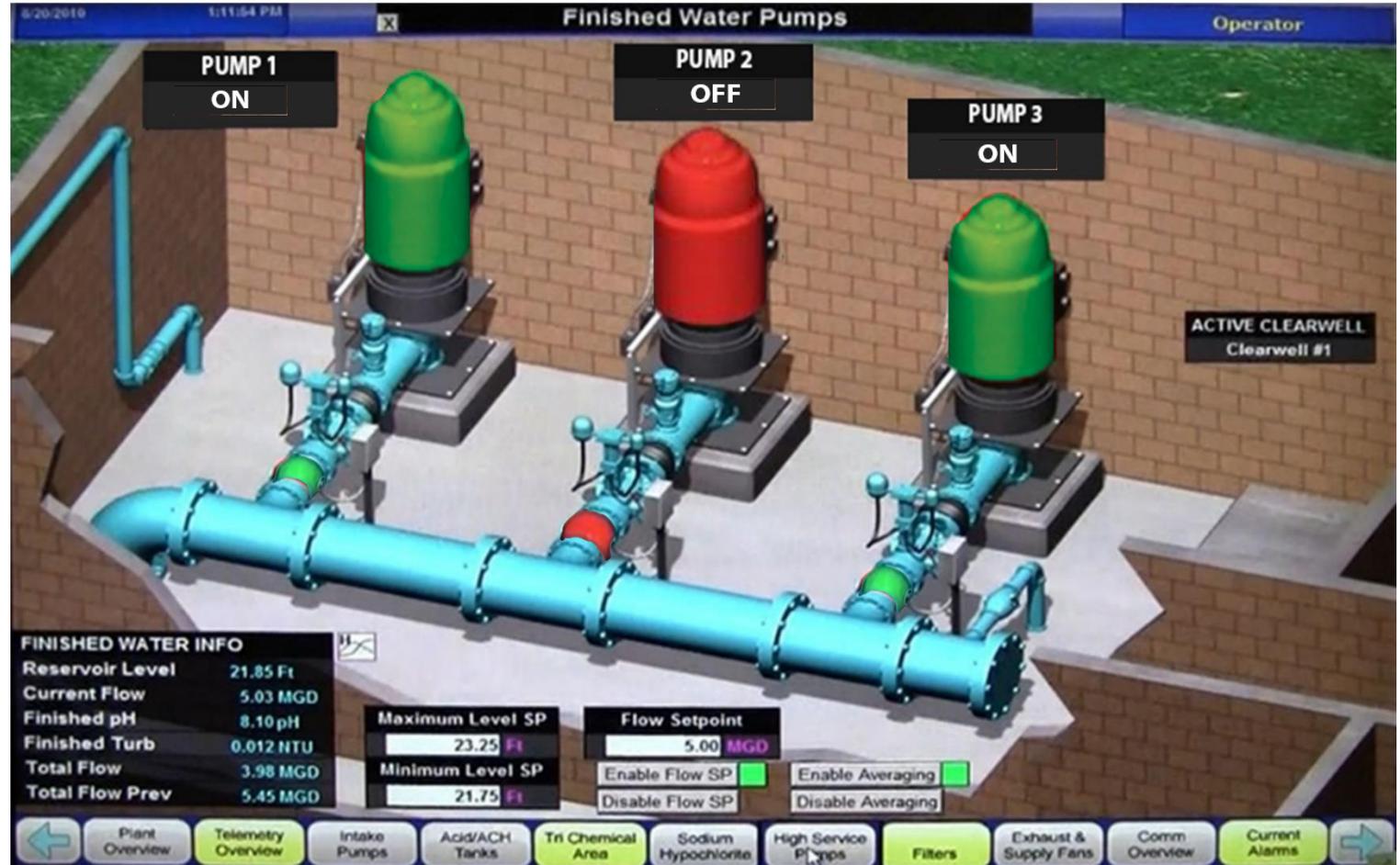
AUTOMATION SYSTEM

Variable-Frequency Drive (VFD) / DC Drive



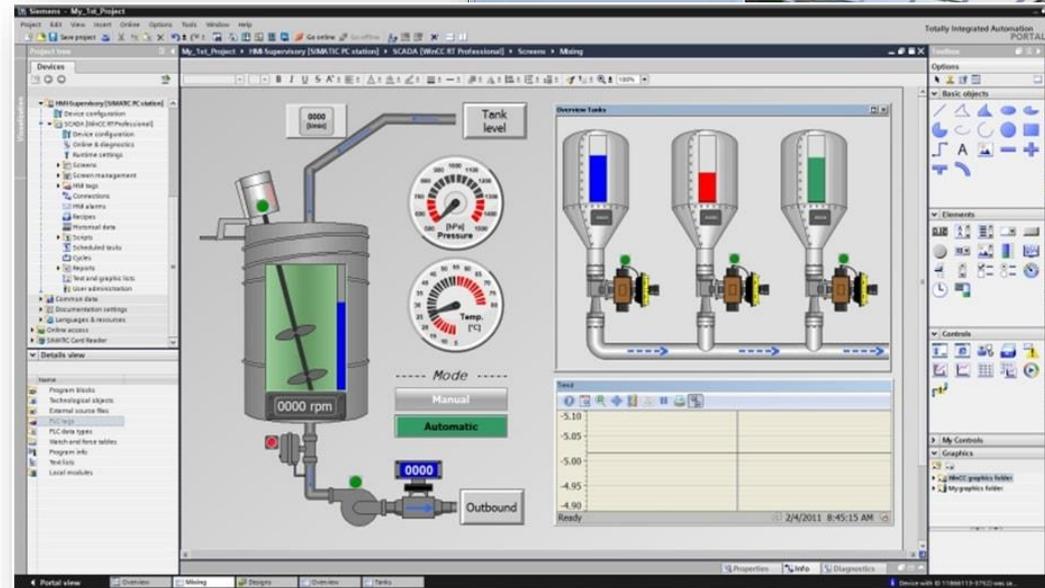
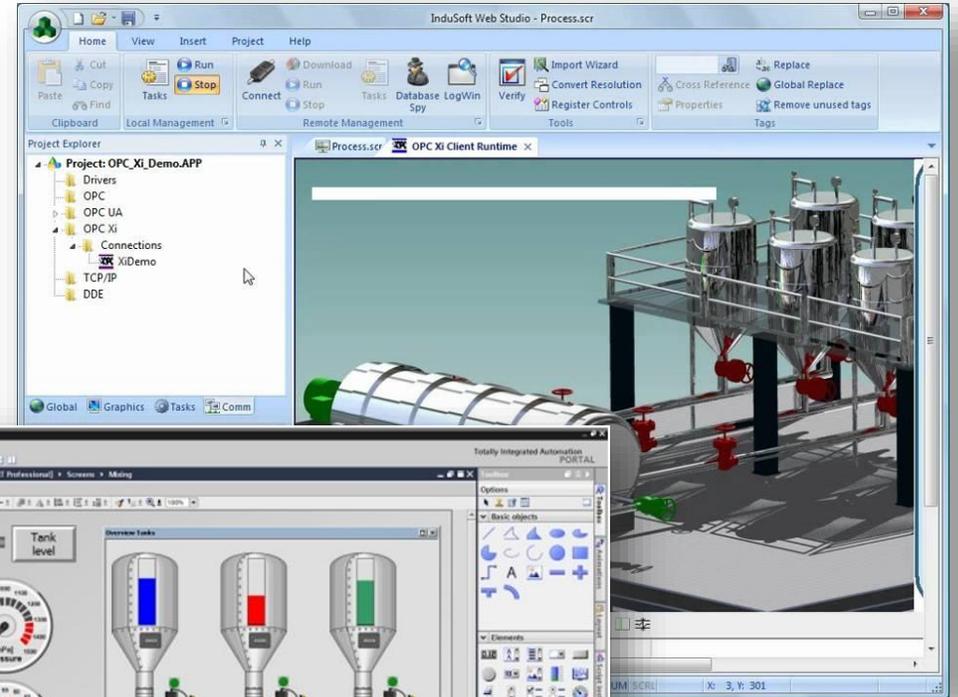
AUTOMATION SYSTEM

Supervisory Control and Data Acquisition (SCADA):



AUTOMATION SYSTEM

Supervisory Control and Data Acquisition (SCADA):



AUTOMATION SYSTEM

INSTRUMENTATION:

Flow transmitter

Pressure transmitter

Temperature Sensor

Level Sensors



AUTOMATION SYSTEM

Light Sensor

Smoke and Gas Sensors

IR Sensor

Proximity Sensors

Sound Level Transmitter

Vibration Sensor



PLC Programming Languages.

The 5 most popular PLC Programming Languages are:

Graphical language

1. **Ladder Diagrams (LD)**
2. **Function Block Diagram (FBD)**
3. **Sequential Function Chart (SFC)**

Textual language

4. **Instructions List (IL)**
5. **Structured Text (ST)**

1. Ladder Diagram (LD):

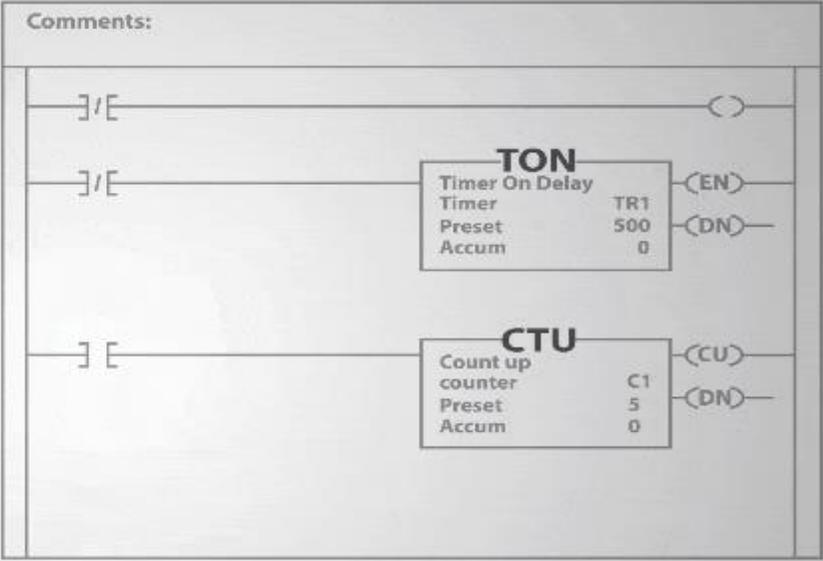
Ladder Diagram was originally modeled from **relay-logic** which used physical devices, such as switches and mechanical relays to control processes. Ladder Diagram utilizes internal logic to replace all, except the physical devices that need an electrical signal to activate them.

1. Ladder Diagram



Relay-Logic Panel

Comments:

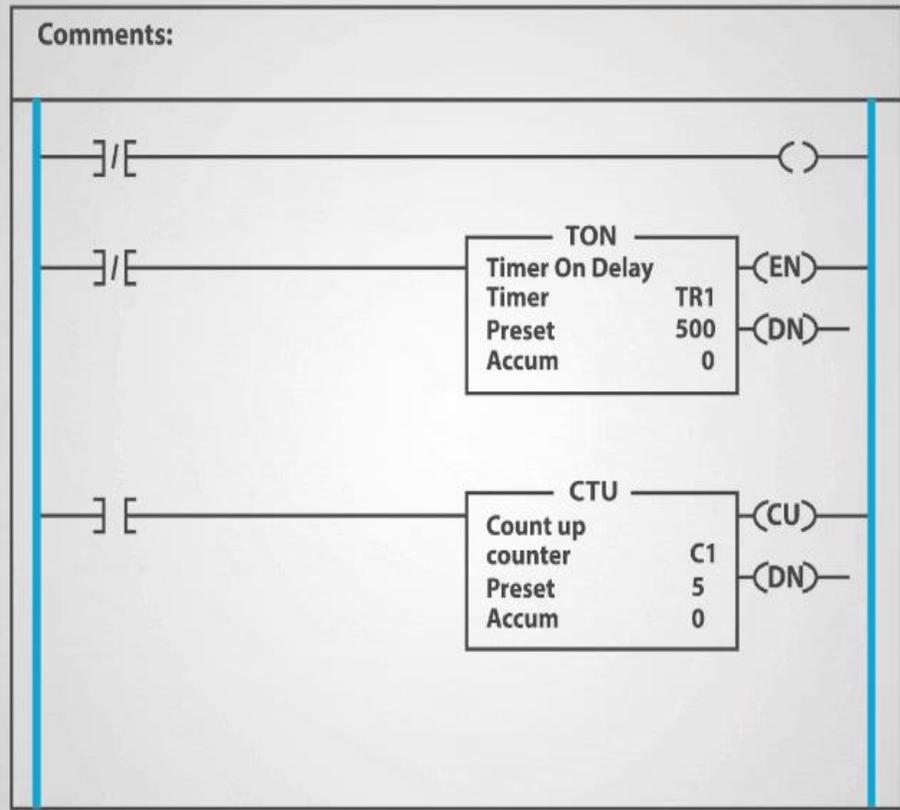


Ladder Diagram

REALPARS

Component	Parameter	Value
TON (Timer On Delay)	Timer	TR1
	Preset	500
	Accum	0
CTU (Count up counter)	Counter	C1
	Preset	5
	Accum	0

1. Ladder Diagram



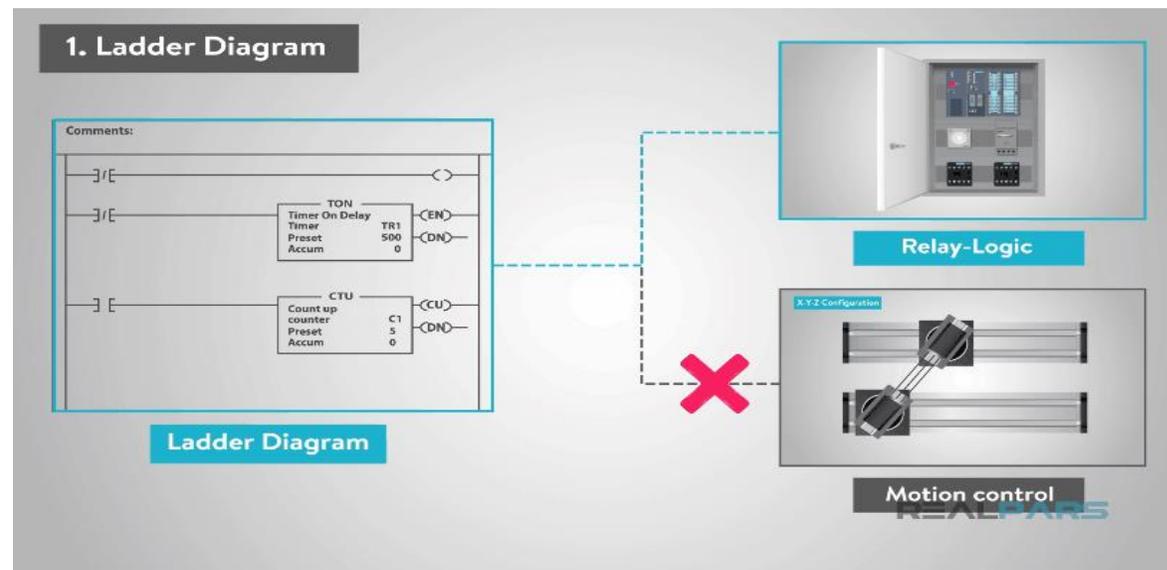
Ladder Diagram

The main advantages of the Ladder Diagram language are:

- 1. The rungs allow it to be organized and easy to follow.**
- 2. It also lets you document comments that are readily visible.**
- 3. It supports online editing very successfully.**

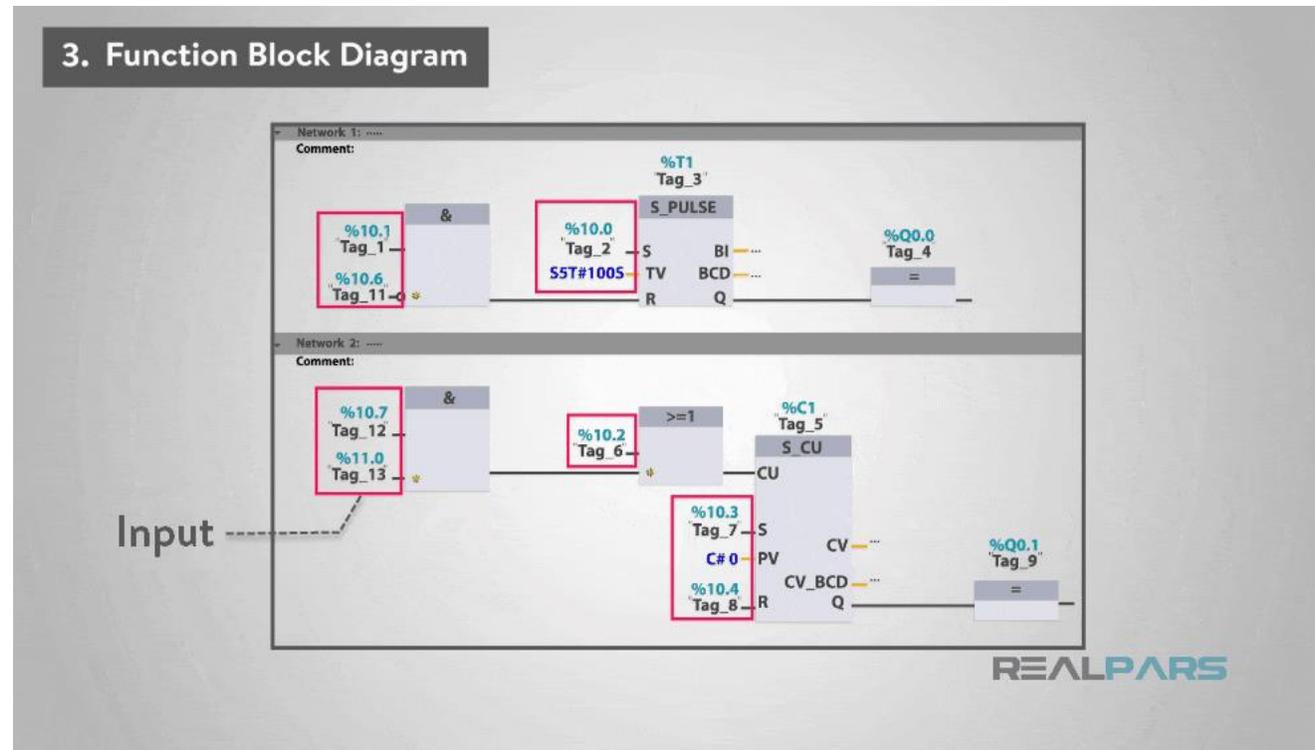
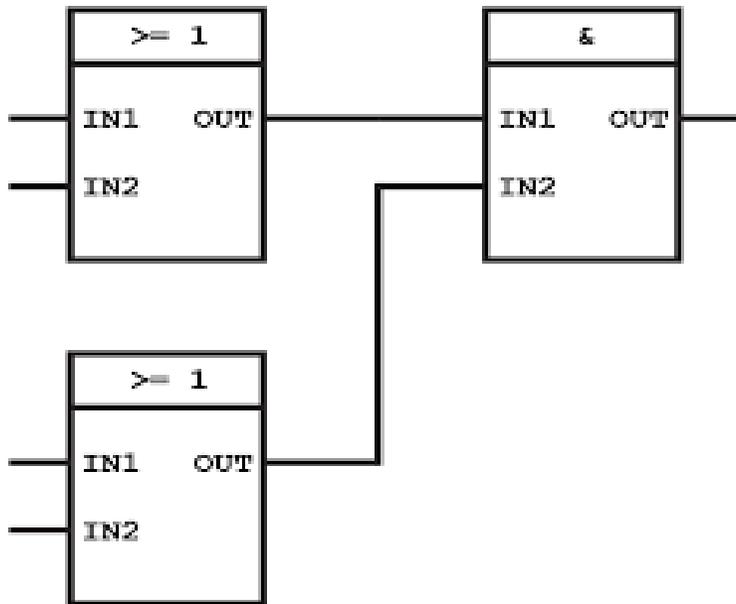
Ladder Diagram Disadvantages:

The main disadvantage is that there are some instructions that are not available, which might make it more difficult for programming such as **motion or batching**.



2. Function Block Diagram (FBD):

The Function Block Diagram (FBD) is a **graphical language** for programmable logic controller design, that can describe the function between input variables and output variables. A function is described as a set of elementary blocks. Input and output variables are connected to blocks by connection lines.



Function Block Diagram Advantages

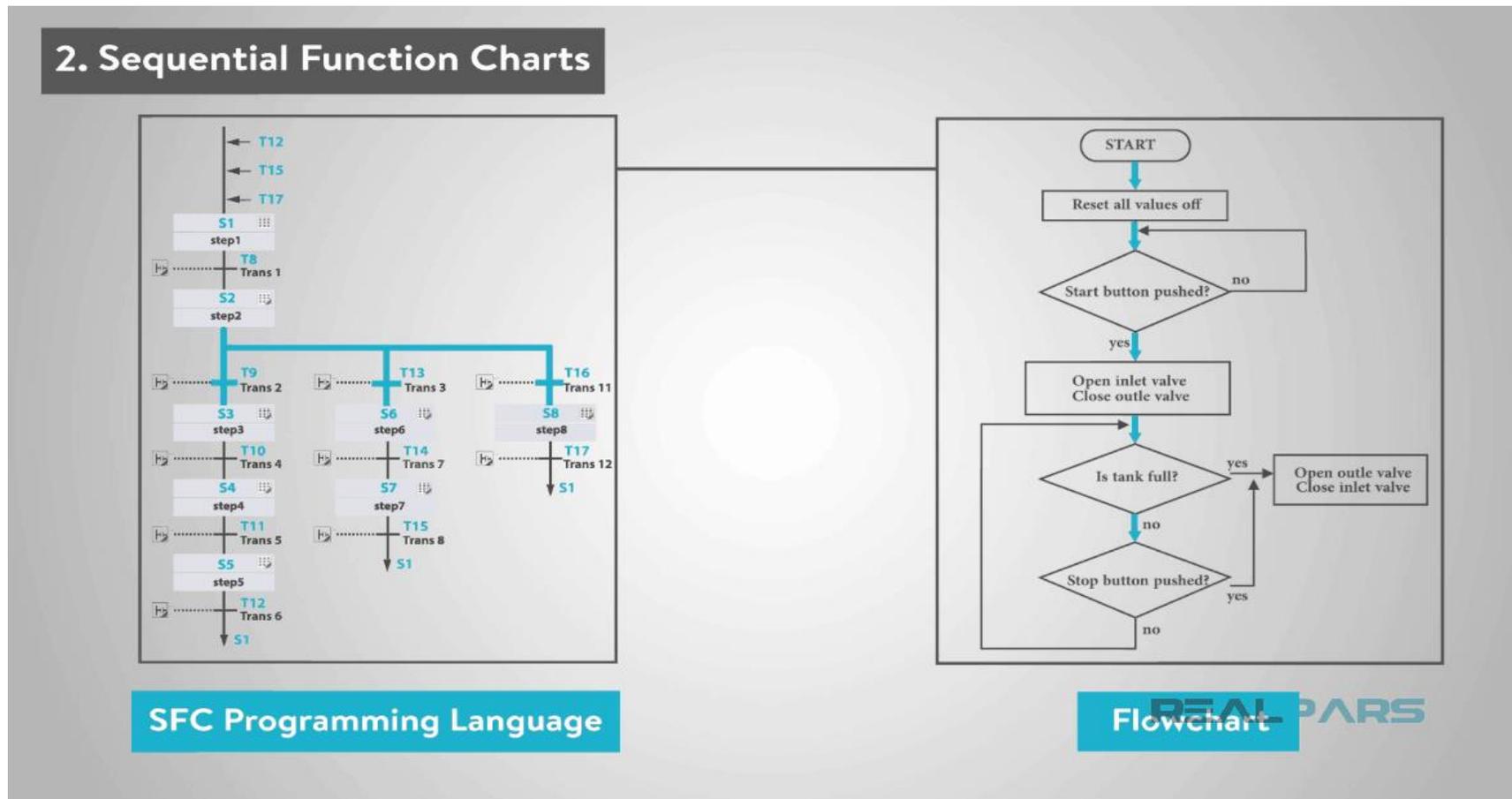
1. The Function Block Diagram does work well with motion controls.
2. The visual method is easier for some users.
3. The biggest advantage of Function Block Diagram is that you can take many lines of programming and put it into one or several function blocks.

Function Block Diagram Disadvantages

The code can get **disorganized** using this PLC Programming Language because you can place the function blocks anywhere on the sheet. This can also make it more difficult to troubleshoot

3. Sequential function charts (or SFCs):

SFCs are a graphical programming language, not text-based. Being a visual programming language means that it's well suited to the task of breaking down a large and complex process into smaller pieces that are easier to see and understand than with solely text-based programming environments.



Sequential Function Charts Advantages:

1. Processes can be broken into major steps that can make troubleshooting faster and easier.
2. Direct access in the logic to see where a piece of equipment faulted.
3. It can be faster to design and write the logic due to the ability to use repeated executions of individual pieces of logic.

Sequential Function Charts disadvantages:

The main disadvantage is that there are some instructions that are not available, which might make it more difficult for programming such as motion or batching.

4. Instruction List (IL):

An instruction list (IL) consists of a series of instructions. Each instruction begins in a new line and contains an operator and depending on the type of operation, one or more operands separated by commas. In front of an instruction there can be an identification mark (label) followed by a colon (:).

5. Instruction List

Instructions

```
1 PROGRAM
2 VAR
3     Timer1   : TON;
4     Timer2   : TON;
5     Tag-1    : BOOL;
6     Tag-2    : INT;
7     tIn1     : TIME;
8     tOut1    : TIME;
9 END_VAR
1 LD Tag-1
2 ST Timer1.IN
3 GOTO mark1
4 CAL Timer1 (
5     PT:=tIn1.
6     ET:=>tout1)
7 LD Timer1.Q
8 ST Timer2.IN
9
10 mark1:
11 LD Tag-2
12 AND 230
13 OR 3
```

Comments

starts timer with rising edge, resets time...

gets TRUE, delay time (PT) after a rising... starts timer with rising edge, resets time...

Applications of IL language

- 1 Compact Coding
- 2 Time Critical Coding

REALPARS

The diagram illustrates the structure of an Instruction List (IL) in a ladder logic programming language. It shows a sequence of instructions starting with a 'PROGRAM' declaration, followed by variable declarations (VAR) for timers and tags. The main body of the IL consists of instructions like LD, ST, GOTO, and CAL, which are used to set up and execute timer operations. Comments are provided for some instructions to explain their function. The diagram also highlights the applications of IL language, such as compact coding and time-critical coding.

Instruction List Advantages

The Instruction List language is valuable for **applications that need code** that is compact and time critical.

Instruction List Disadvantages

1. There are few structuring possibilities with the “Goto” command being one of them.
2. **There can also be many errors that are more difficult to deal** with in comparison to many of the other languages that I have previously reviewed.

5. Instruction List

1	PROGRAM		
2	VAR		
3	Timer1	: TON;	
4	Timer2	: TON;	
5	Tag-1	: BOOL;	
6	Tag-2	: INT;	
7	tIn1	: TIME ;	
8	tOut1	: TIME ;	
9	END_VAR		
1	LD	Tag-1	
	ST	Timer1.IN	starts timer with rising edge, resets time...
	GOTO	mark1	
	CAL	Timer1 (
		PT:=tIn1.	
		ET:=>tout1)	
	LD	Timer1.Q	gets TRUE, delay time (PT) after a rising...
	ST	Timer2 . IN	starts timer with rising edge, resets time...
2	mark1:		
	LD	Tag-2	
	AND	230	
	OR	3	



LIST OF ERRORS

- Fatal Error
- Run-time Error
- Odd Address Error
- Programming Error

REALPARS

5. Structured Text (ST):

Structured Text language is a **textual based language**. Structured Text is a high-level language that is like Basic, Pascal and “C”. It is a very powerful tool that **can execute complex tasks** utilizing algorithms and mathematical functions along with repetitive tasks. The code uses statements that are separated by semicolons and then either inputs, outputs, or variables are changed by these statements.

it uses functions such as FOR, WHILE, IF, ELSE, ELSEIF AND CASE.

4. Structured Text

```
FOR WHILE IF ELSE ELSE IF CASE ...
1 IF #Enable = 1 THEN
2
3 //Init
4 #The_Max := #In_1;
5 #i := 1;
6 #Array [1] := #In_1;
7 #Array [2] := #In_2;
8 #Array [3] := #In_3;
9 #Array [4] := #In_4;
10 #Array [5] := #In_5;
11 //*****
12 WHILE #i < 6 DO
13 IF #Array[#i] > #The_Max THEN
14 #The_Max := #Array[#i];
15 ELSE
16 #i := #i + 1;
17 END_IF;
18 END_WHILE;
19 END IF;
```

Structured Text Advantages

1. It is very organized and good at computing large mathematical calculations.
2. It will enable you to cover some instructions that are not available in some other languages like the Ladder Diagram.

Structured Text Disadvantages

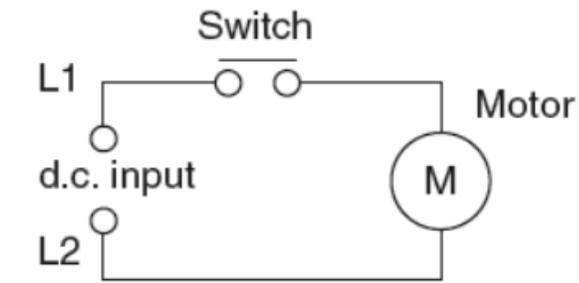
1. The syntax can be difficult.
2. It is hard to debug.
3. It is difficult to edit online.

the Ladder Diagram is by far the most popular PLC programming language .

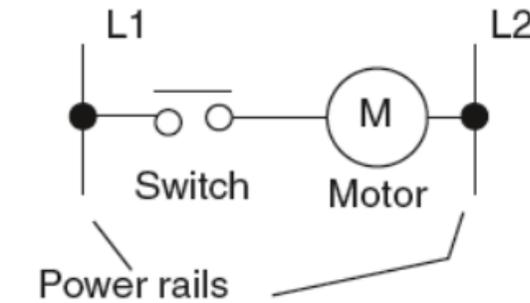
The main reason for this is that the Ladder Diagram language naturally followed the technology advancement from **a physical relay logic to a digital and logical one**. This allowed the engineers and skilled workers to follow and troubleshoot and make that transition.

Ladder Diagrams

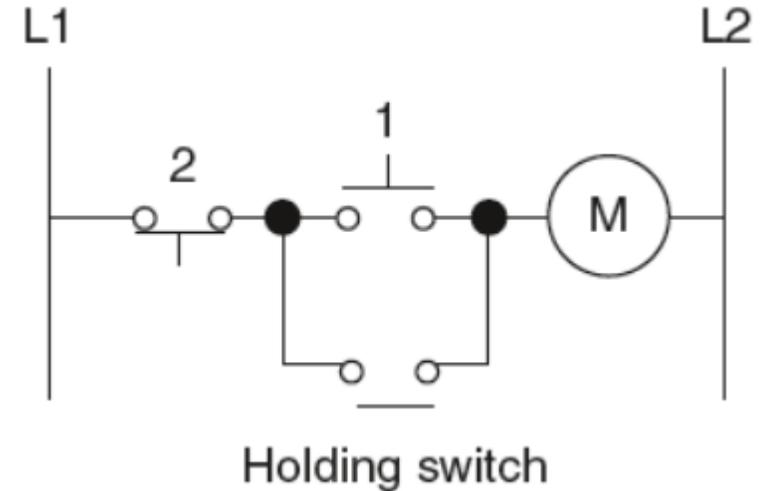
As an introduction to ladder diagrams, consider the simple wiring diagram for an electrical circuit in Figure. The diagram shows the circuit for switching on or off an electric motor. We can redraw this diagram in a different way, using two vertical lines to represent the input power rails and stringing the rest of the circuit between them.



(a)

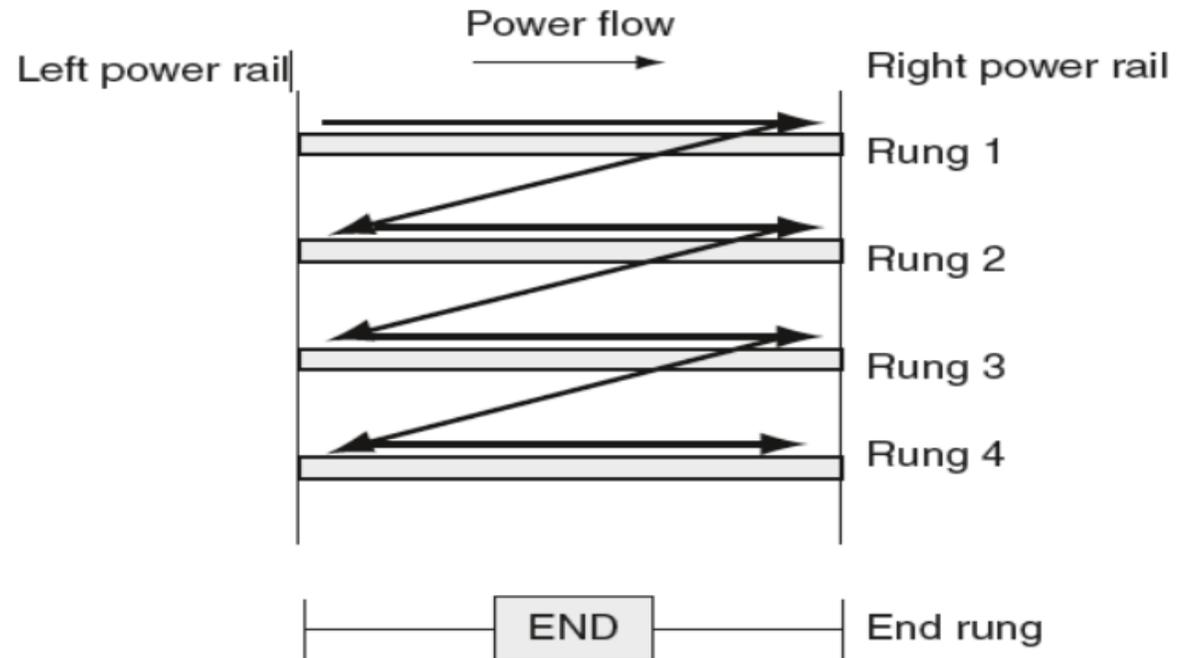


(b)



PLC Ladder Programming:

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.

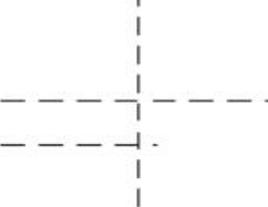
3. A ladder diagram is read from left to right and from top to bottom, Figure showing the scanning motion employed by the PLC. The top rung is read from left to right. Then the second rung down is read from left to right and so on. When the PLC is in its run mode, it goes through the entire ladder program to the end, the end rung of the program being clearly denoted, and then promptly resumes at the start. 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.

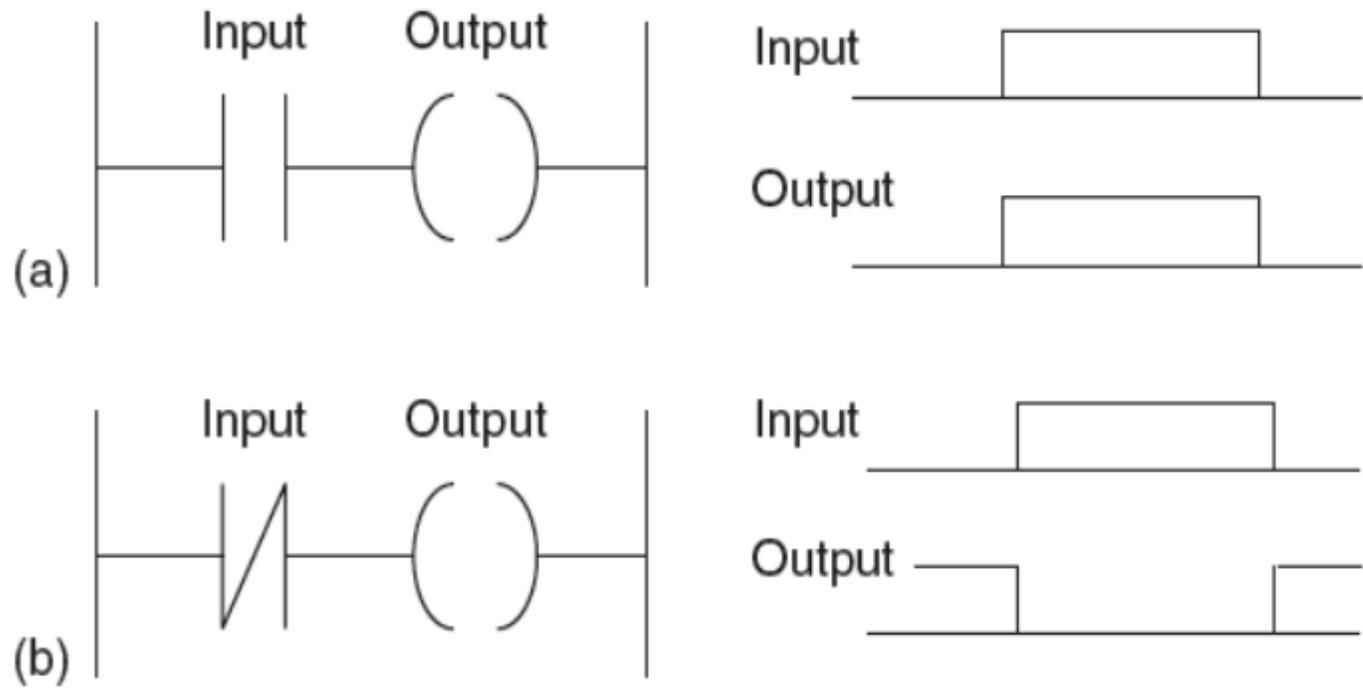
4. Each rung must start with an input or inputs and must end with at least one output. The term input is used for a control action, such as closing the contacts of a switch, used as an input to the PLC. The term output is used for a device connected to the output of a PLC, e.g., a motor.

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.

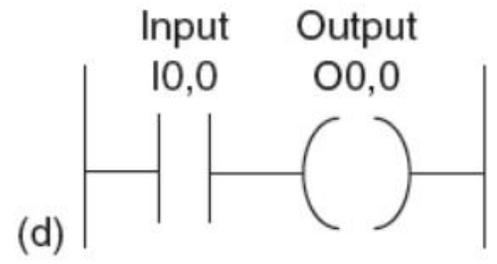
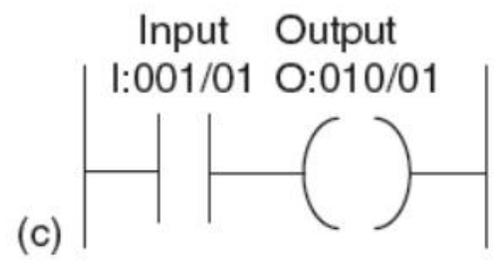
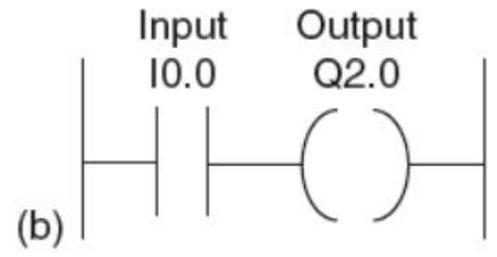
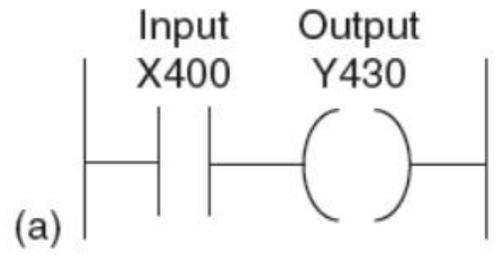
6. A particular device can appear in more than one rung of a ladder. For example, we might have a relay that switches on one or more devices. The same letters and/or numbers are used to label the device in each situation.

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.

	<i>Semi-graphic form</i>	<i>Full graphic form</i>
A horizontal link along which power can flow		
Interconnection of horizontal and vertical power flows		
Left-hand power connection of a ladder rung		
Right hand power connection of a ladder rung		
Normally open contact		
Normally closed contact		
Output coil: if the power flow to it is on then the coil state is on		



A ladder rung

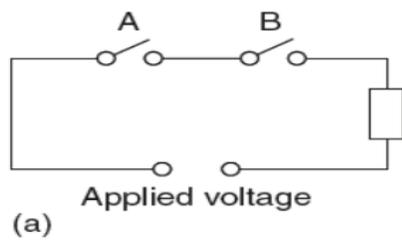
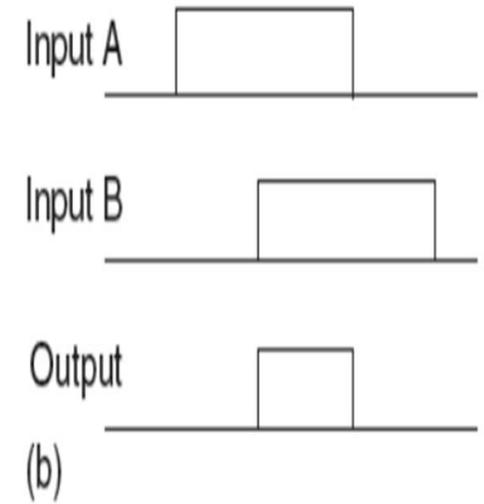
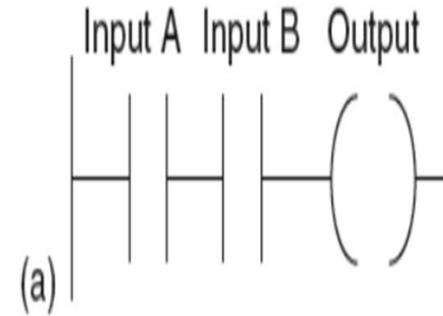


Notation: (a) Mitsubishi (b) Siemens (c) Allen-Bradley (d) Telemecanique

Logic Functions:

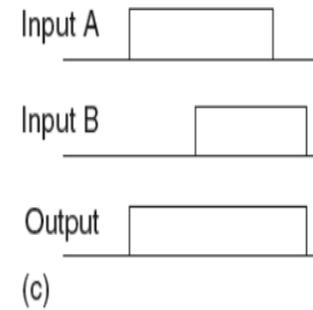
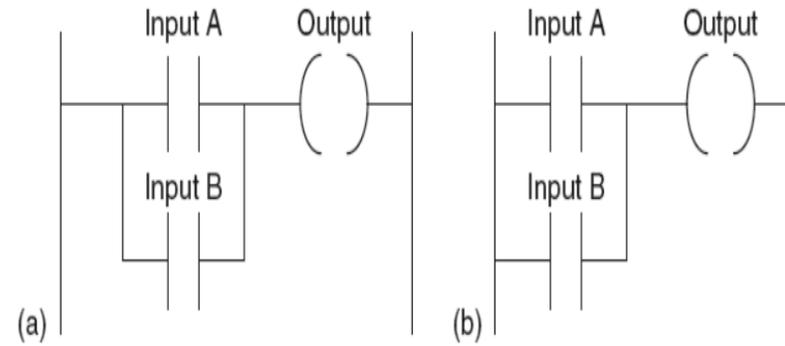
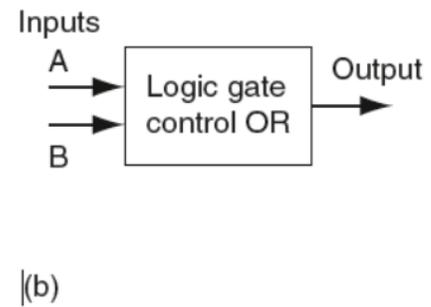
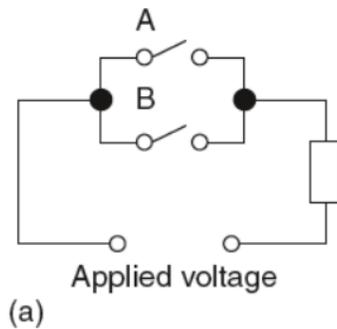
AND logic

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

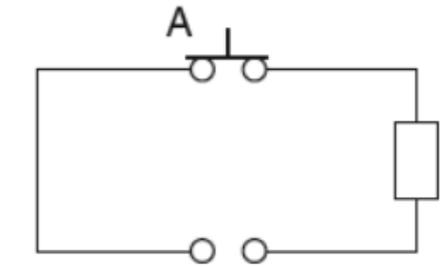


OR gate

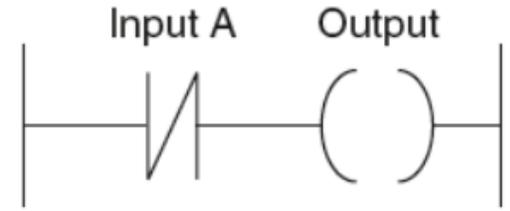
Inputs		Output
A	B	
0	0	0
0	1	1
1	0	1
1	1	1



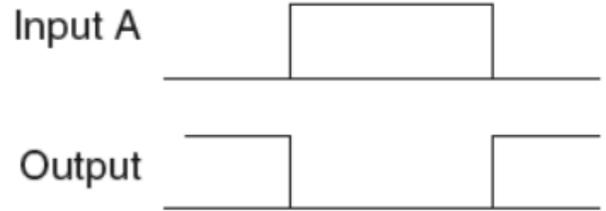
NOT circuit



(a) Applied voltage



(b)

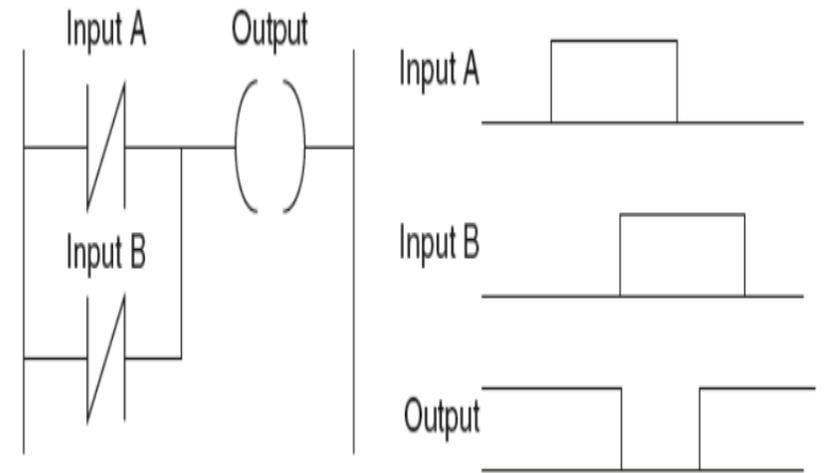
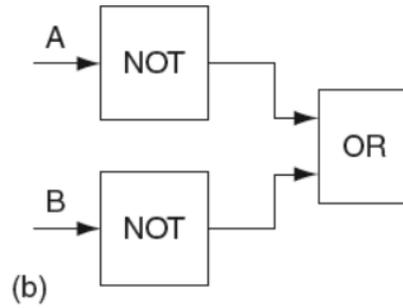
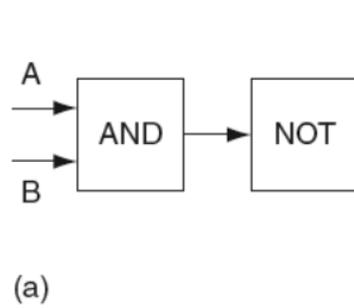


(c)

(a) NOT circuit (b) NOT logic with a ladder rung (c) high output when no input to A

NAND gate

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



NOR gate

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

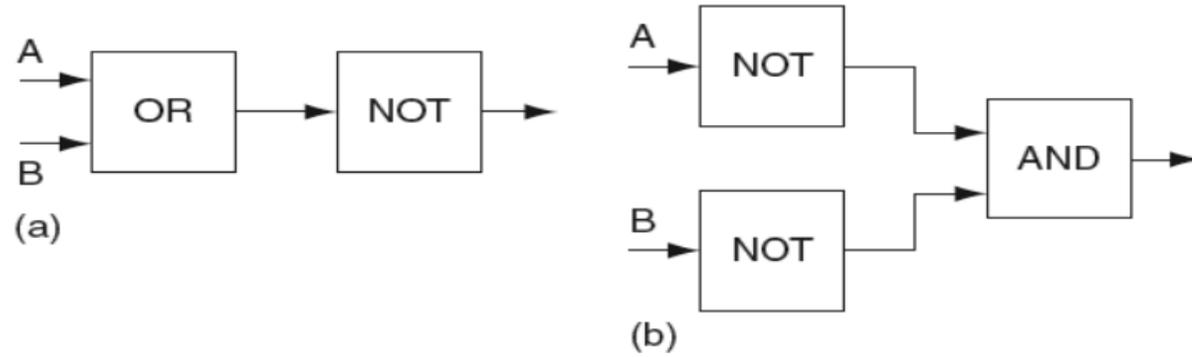
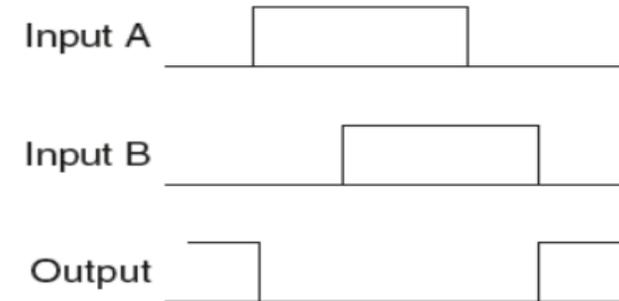
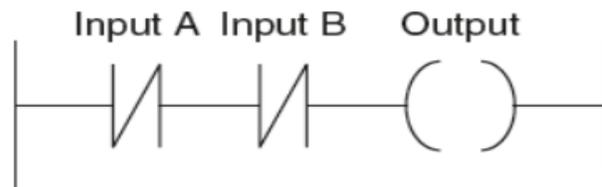
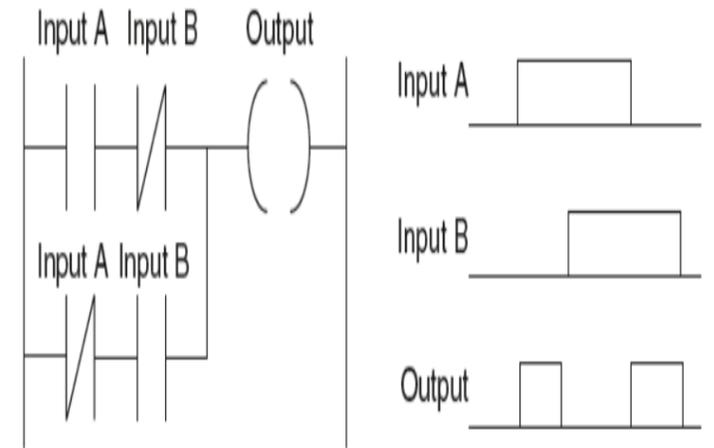
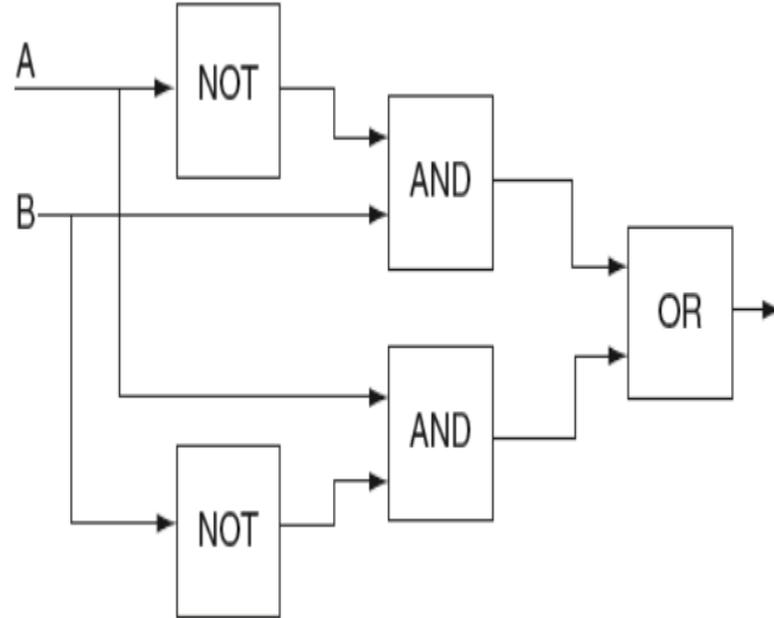


Figure 11.14: NOR gate



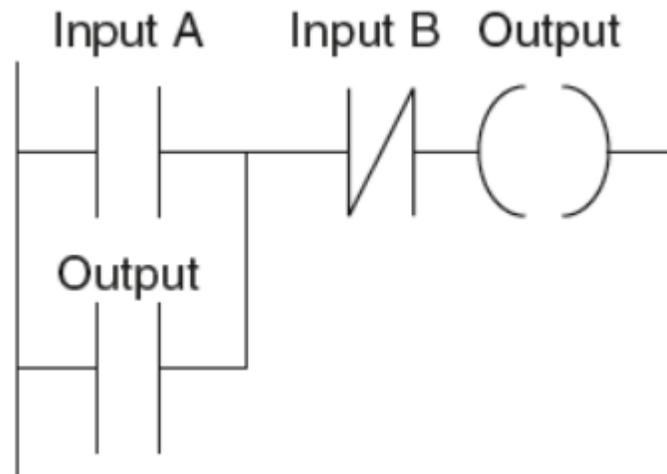
XOR gate

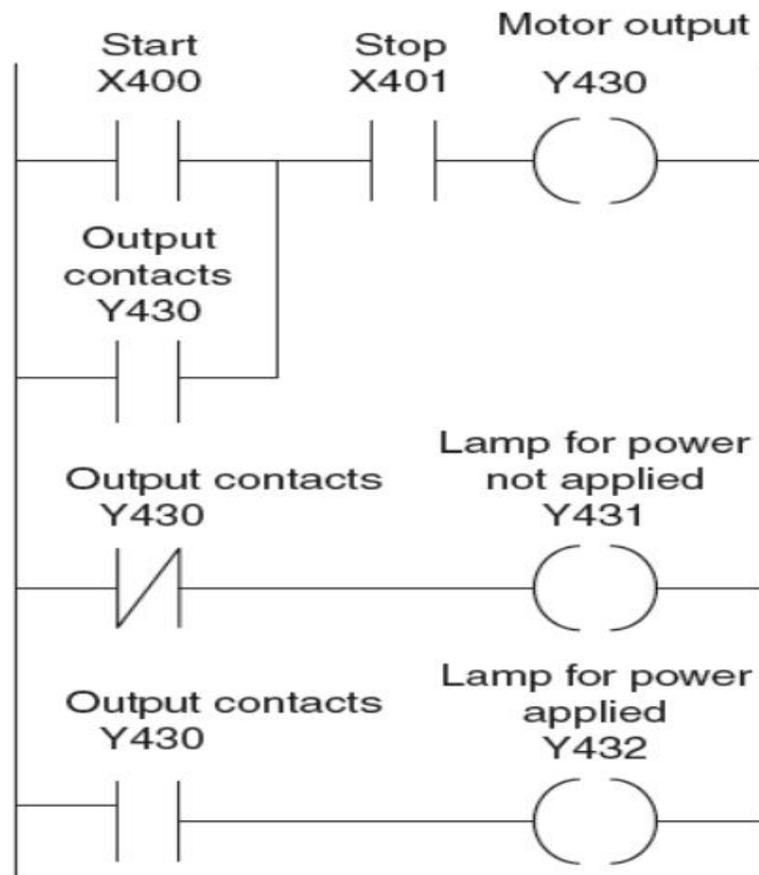
Inputs		Output
A	B	
0	0	0
0	1	1
1	0	1
1	1	0



Latching

There are often situations where it is necessary **to hold an output energized**, 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 energized, **it maintains that state until another input is received**.



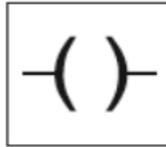


Motor on-off, with signal lamps, ladder diagram.

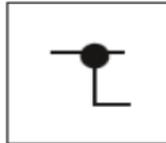
Ladder Symbols



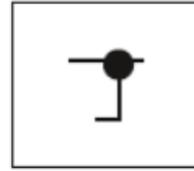
might be used, followed by its address being keyed in. To enter an output the key marked:



might be used, followed by its address. To indicate the start of a junction:



might be pressed; to indicate the end of a junction path:



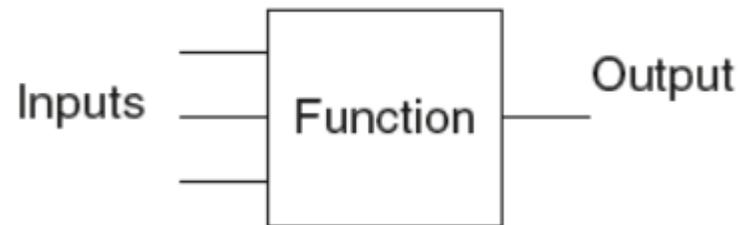
To indicate horizontal circuit links, the following key might be used:

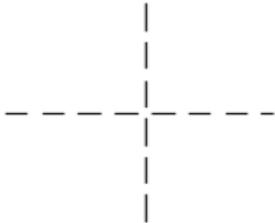
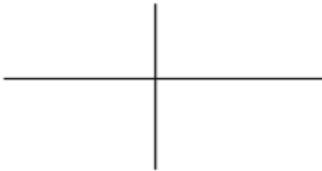
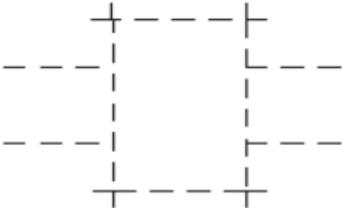
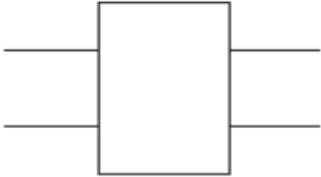


The terminal then translates the program drawn on the screen into machine language.

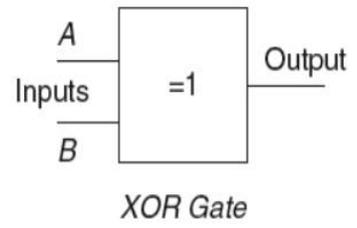
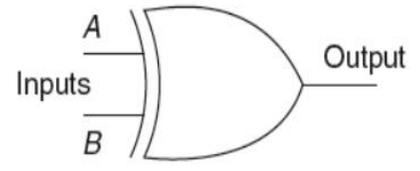
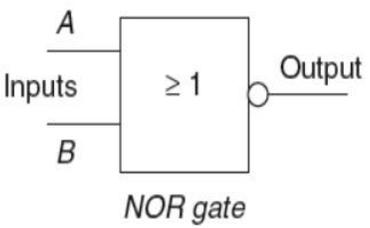
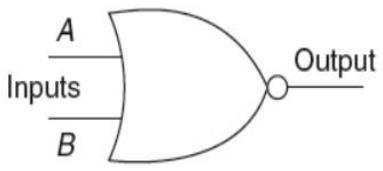
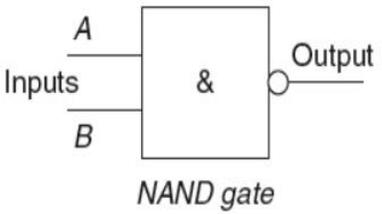
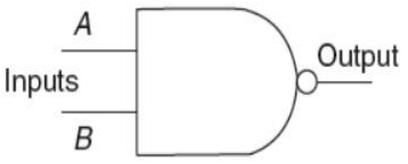
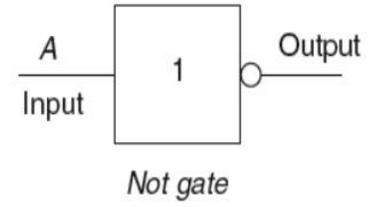
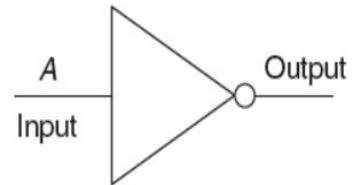
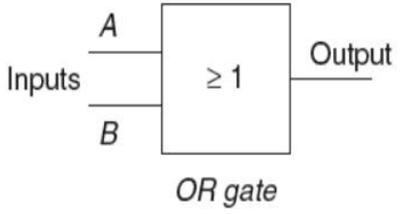
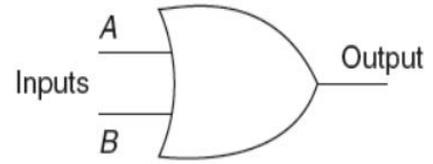
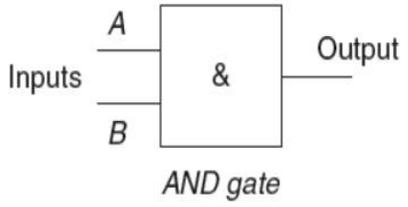
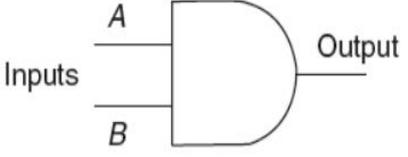
Function Blocks

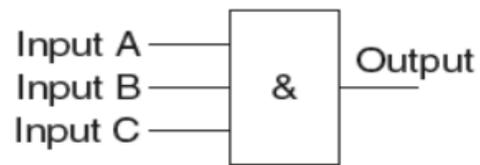
The term function block diagram (FBD) is used for PLC programs described in terms of graphical blocks. It is described as being a graphical language for depicting signal and data flows through blocks, these being reusable software elements. A function block is a program instruction unit which, when executed,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



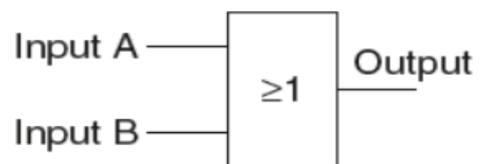
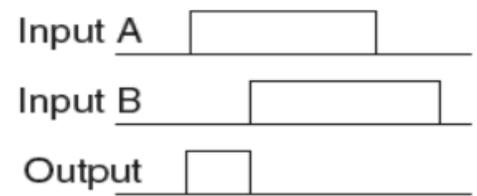
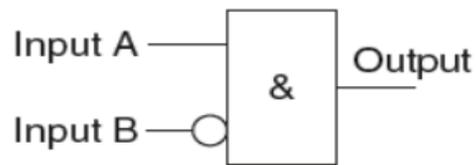
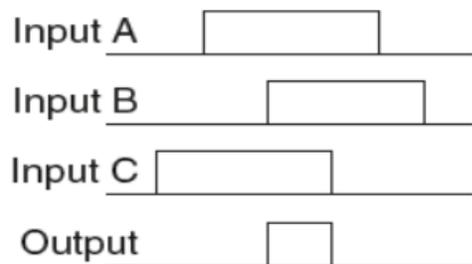
	<i>Semi-graphic form</i>	<i>Full graphic form</i>
Horizontal and vertical lines		
Interconnection of horizontal and vertical signal flows		
Crossing horizontal and vertical signal flows		
Blocks with connections		
Connectors	 	 

Logic Gates

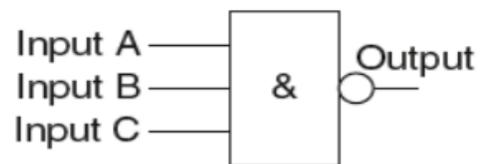
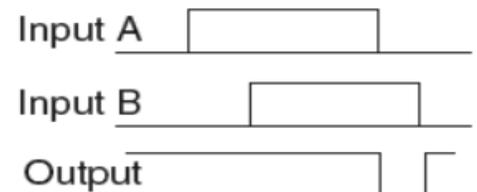
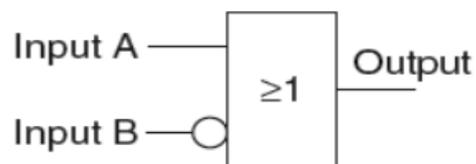
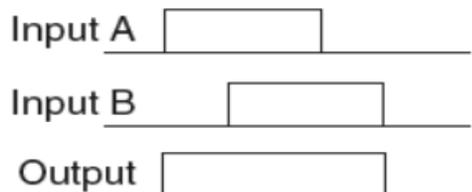




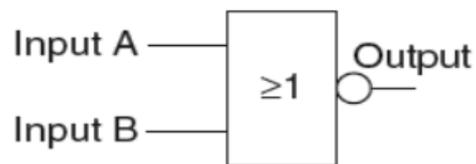
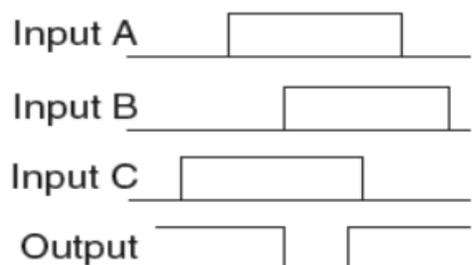
AND function



OR function



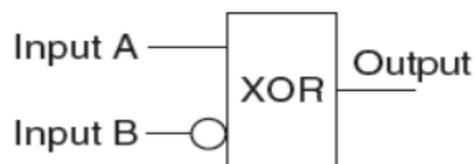
NAND function



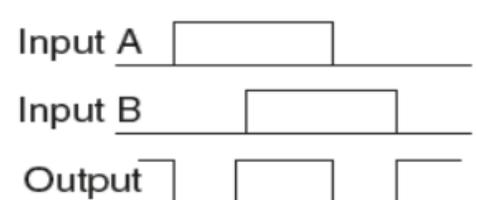
NOR function

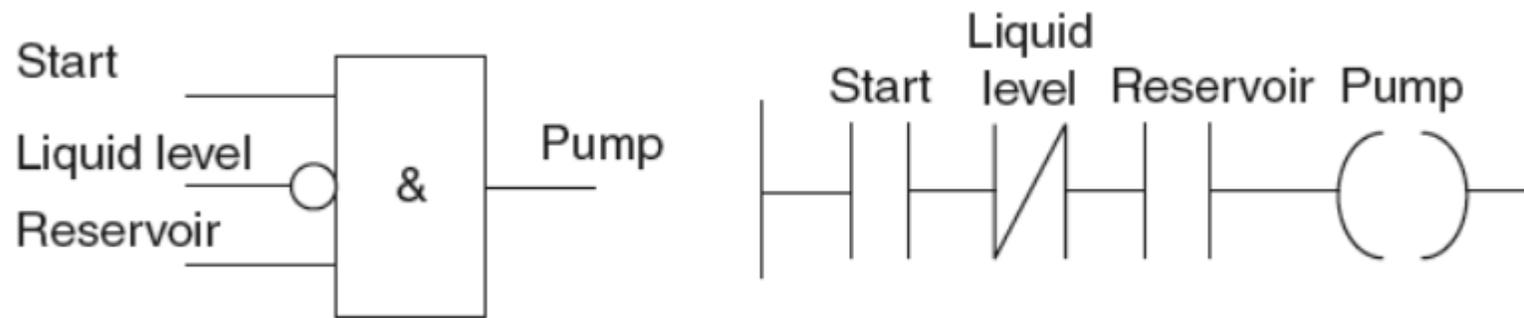


XOR function



XOR function

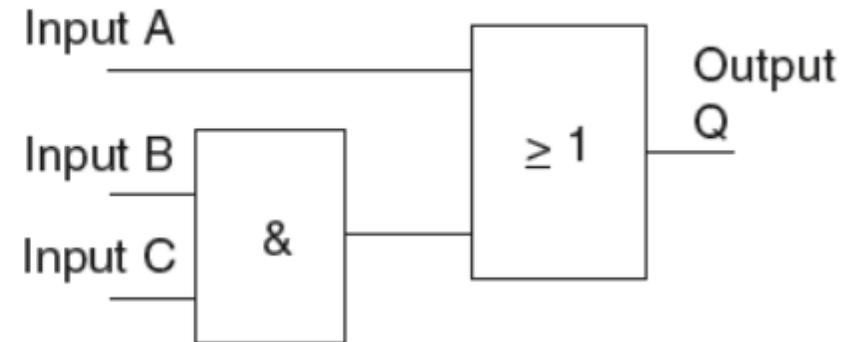
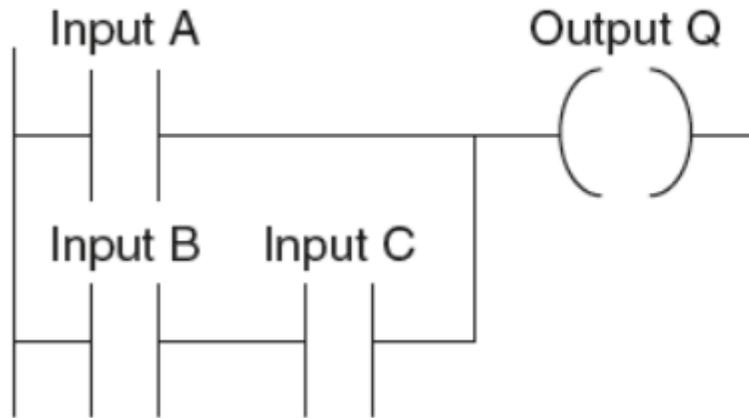


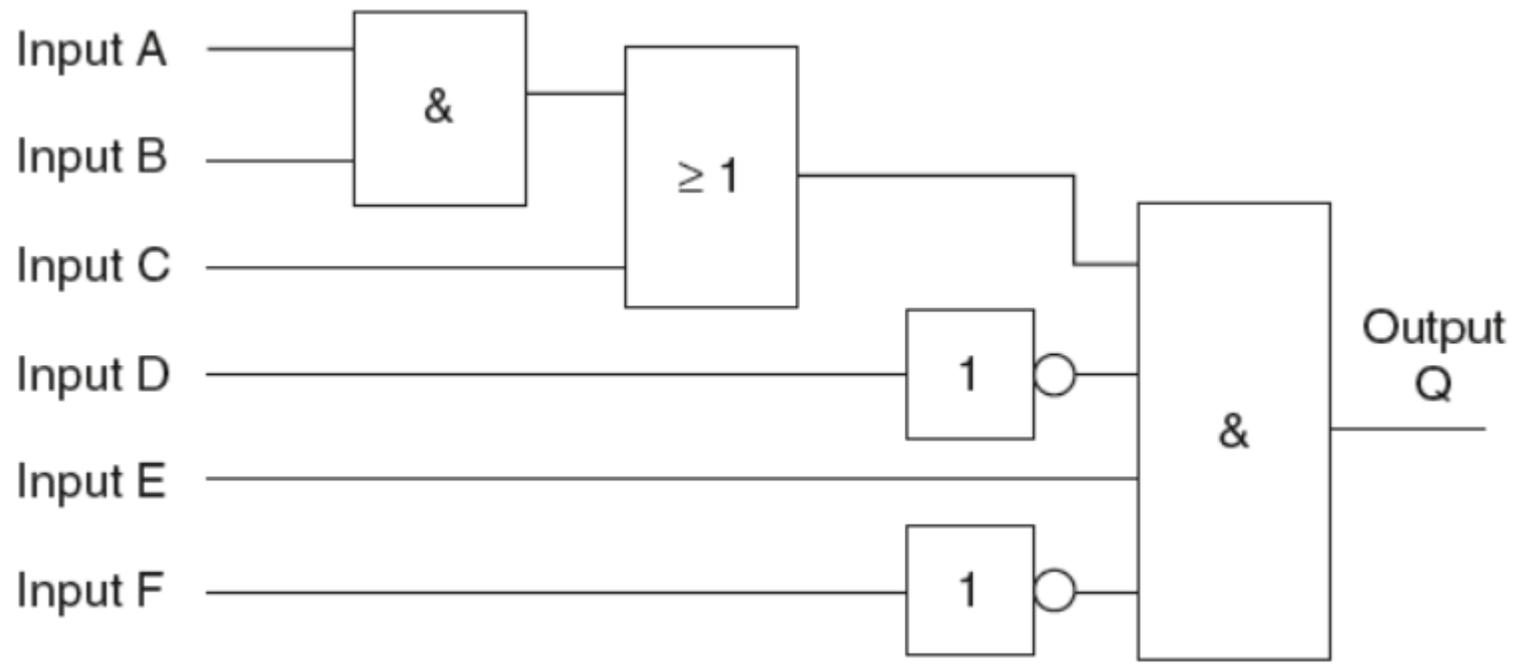


Pump application

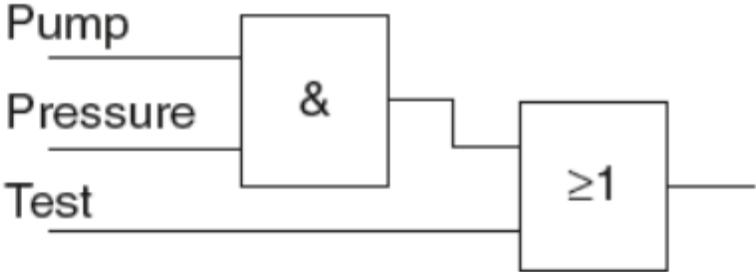
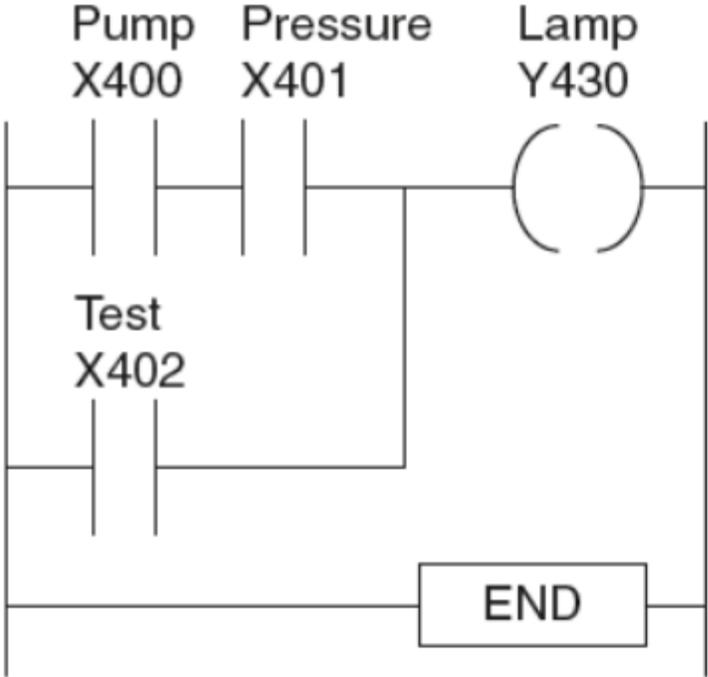
Boolean Algebra:

$$A + B \cdot C = Q$$





Program Examples



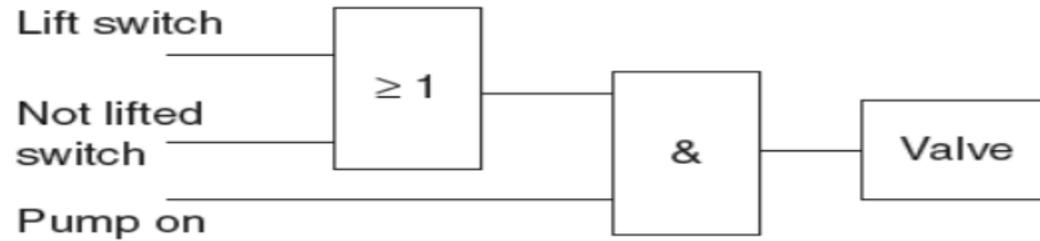
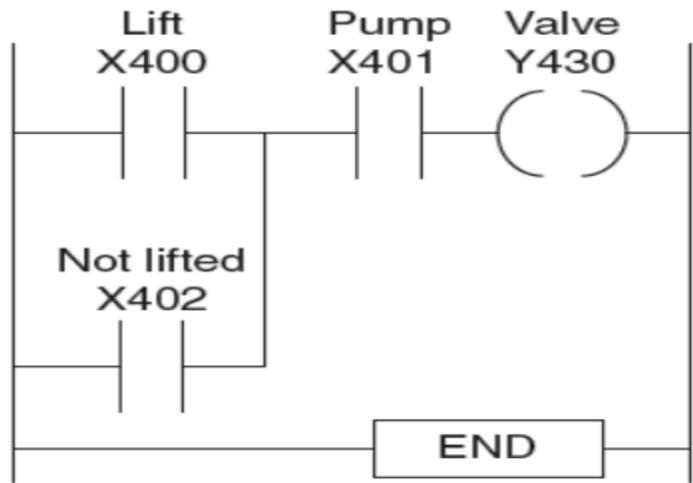


Figure 11.38: Valve operation program

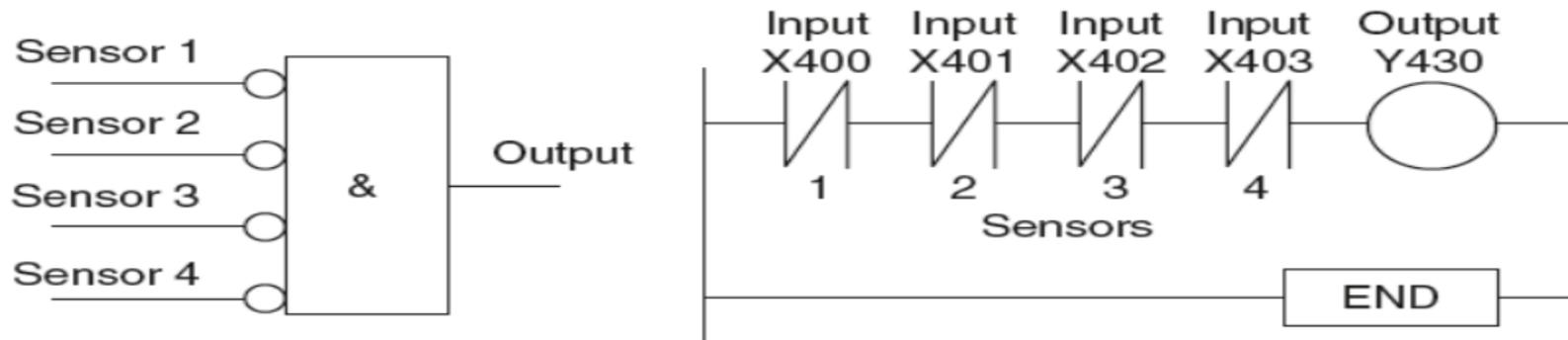
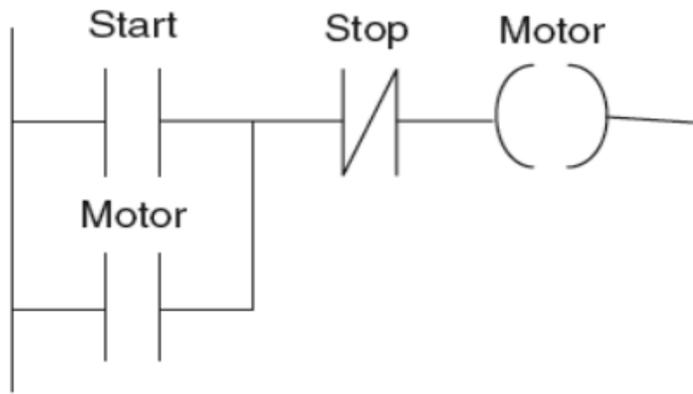
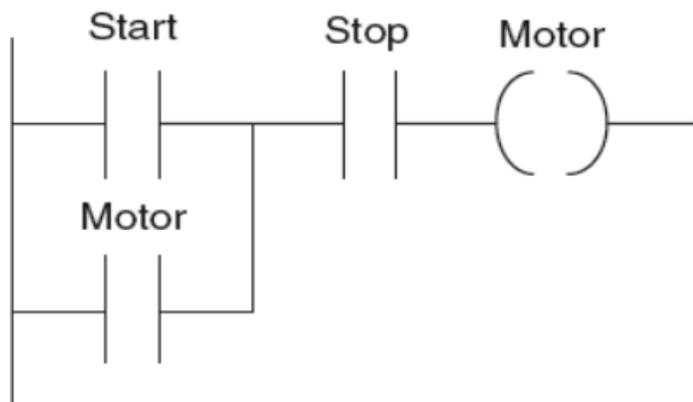
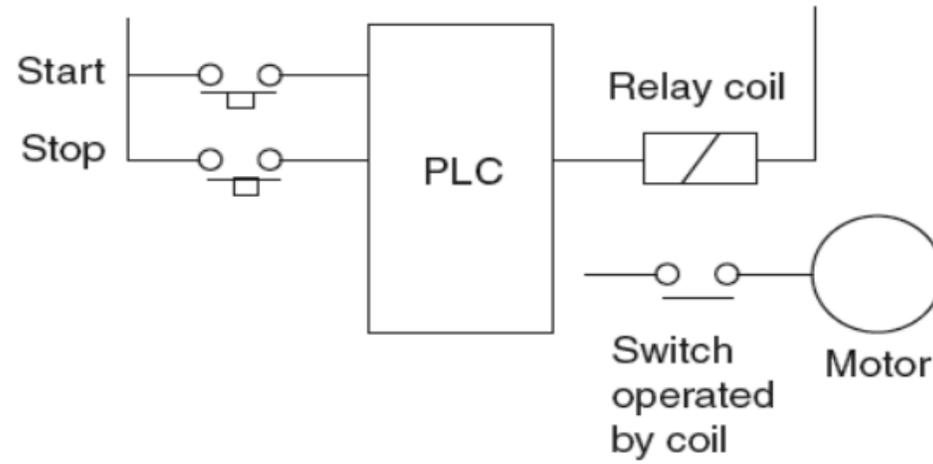


Figure 11.39: Output switched off by any one of four sensors being activated



(a) An unsafe stop switch



(b) A safe stop switch

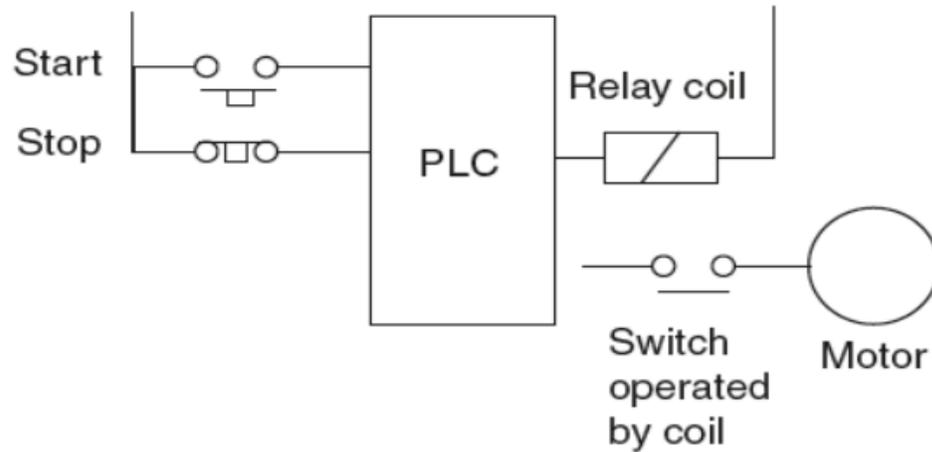


Figure 11.40: Motor stop switch location

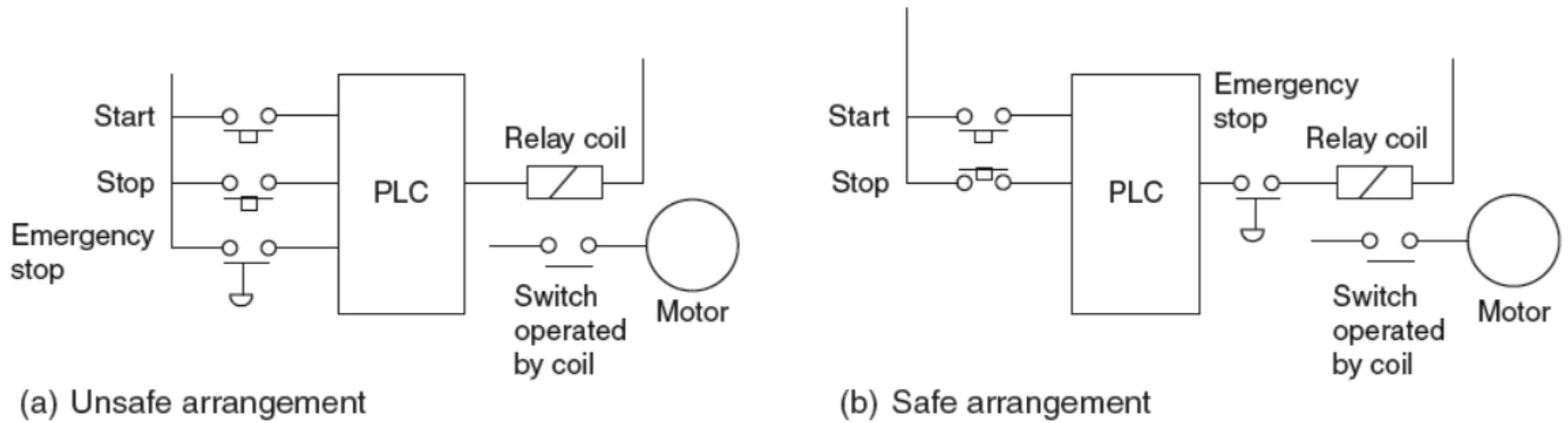


Figure 11.41: Location of emergency stop switch

IL, SFC and ST programming methods

IEC 1131-3 (International Electro technical Commission) programming languages,

1. Instruction lists (IL),
2. Sequential function charts (SFC),
3. Structured text (ST).

A programming method, which can be considered to be the entering of a ladder program using text, is instruction lists (IL). Instruction list gives programs which consist of a series of instructions, each instruction being on a new line. An instruction consists of an operator followed by one or more operands, i.e. the subjects of the operator.

The codes used differ to some extent from manufacturer to manufacturer, though a standard IEC 1131-3 has been proposed and is being widely adopted. Table 1 shows some of the codes used by manufacturers, and the proposed standard, for instructions used in this chapter (see later chapters for codes for other functions).

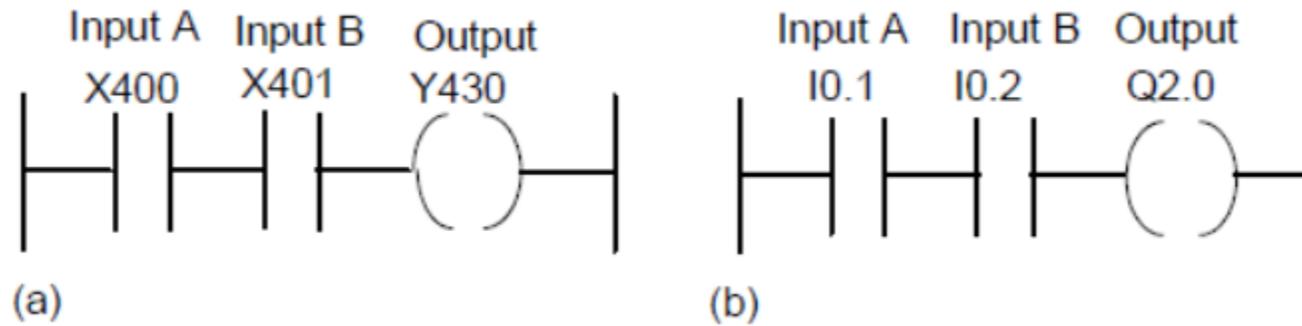
IEC 1131-3 Operators	Mitsubishi	OMRON	Siemens/ Telemecanique	Operation	Ladder diagram
LD	LD	LD	A	Load operand into result register	Start a rung with open contacts
LDN	LDI	LD NOT	AN	Load negative operand into result register	Start a rung with closed contacts
AND	AND	AND	A	Boolean AND	A series element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

As an illustration of the use of IEC 113-1 operators, consider the following:

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

LD is the operator, A the operand, and the words at the ends of program lines and in brackets and preceded and followed by * are comments added to explain what the operation is and are not part of the program operation instructions to the PLC. LD A is thus the instruction to load the A into the memory register. It can then later be called on for further operations. The next line of the program has the Boolean operation AND performed with A and B. The last line has the result stored in Q, i.e. outputted to Q.

Ladder programs and instruction lists



AND gate: (a) Mitsubishi, (b) Siemens

The rung start with LD because it is starting with open contacts. Since the address of the input is X400, the instruction is LD X400. This is followed by another open contacts input and so the next program line involves the instruction AND with the address of the element, thus the instruction is AND X401. The rung terminates with an output and so the instruction OUT is used with the address of the output, i.e. OUT Y430. The single rung of a ladder would thus be entered as:

LD X400

OUT Y430

AND X401

For the same rung with Siemens notation we have:

A I0.1

A I0.2

= Q2.0

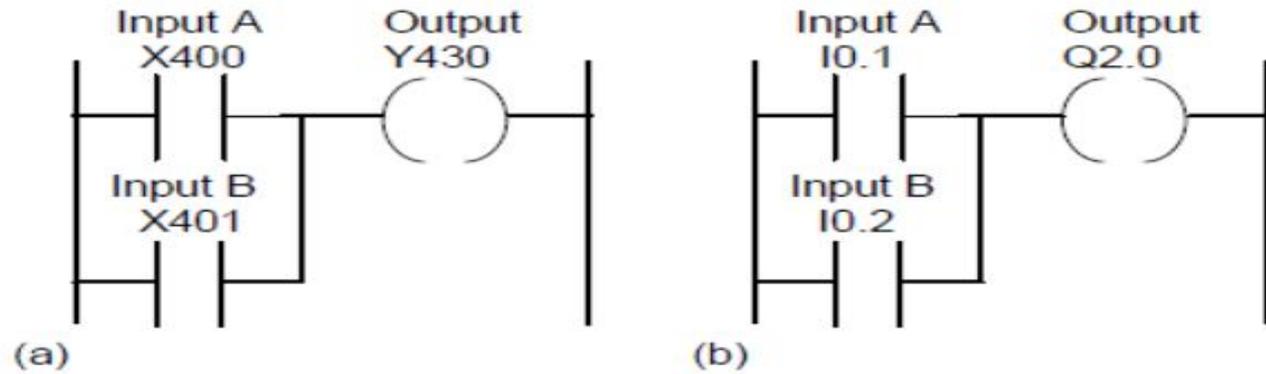


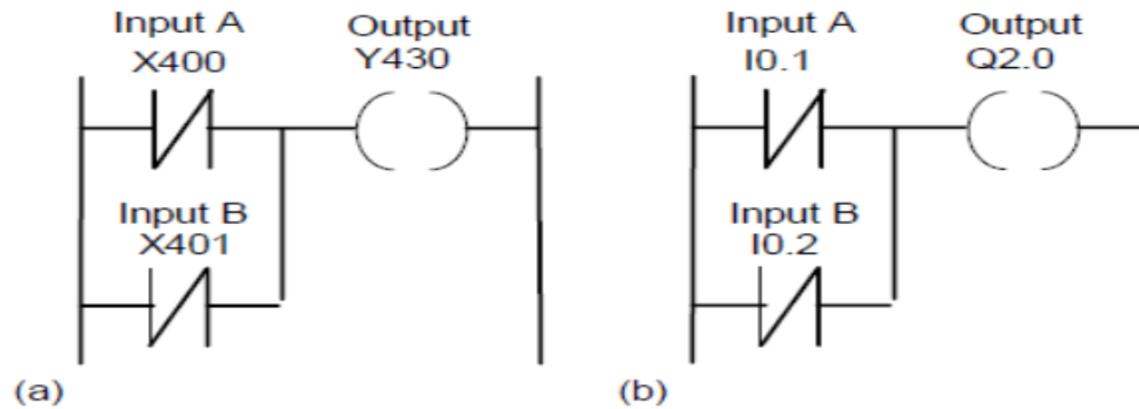
Fig : OR gate: (a) Mitsubishi, (b) Siemens

The instructions for the rung start with an open contact is LD X400. The next item is the parallel OR set of contacts X401. Thus the next instruction is OR X401. The last step is the output, hence OUT Y430. The instruction list would thus be:

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

The Siemens version of the OR gate. The following is the Siemens instruction list:

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



NAND gate: (a) Mitsubishi, (b) Siemens

It starts with the normally closed contacts X400 and so starts with the instruction LDI X400. The next instruction is for a parallel set of normally closed contacts, thus the instruction is ORI X401. The last step is the output, hence OUT Y430. The instruction list is thus:

```
LDI X400
ORI X401
OUT Y430
```

Programming examples

A signal lamp is required to be switched on if a pump is running and the pressure is satisfactory, or if the lamp test switch is closed.

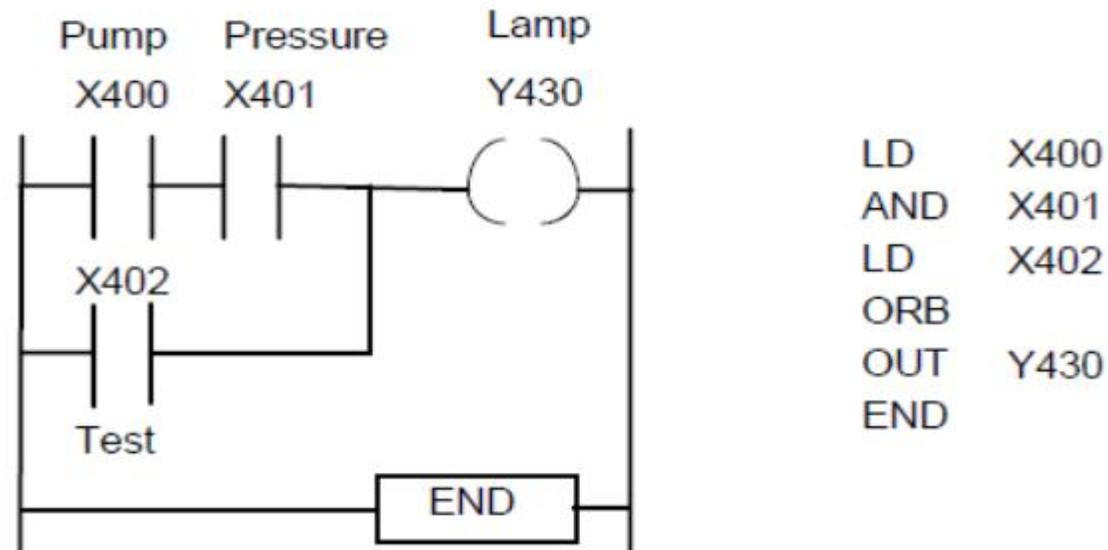
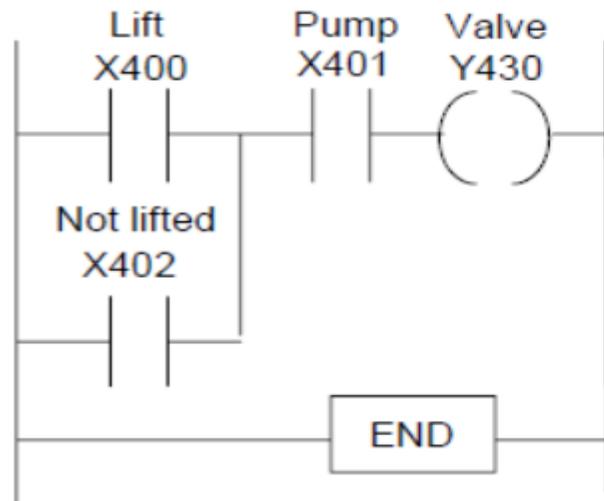
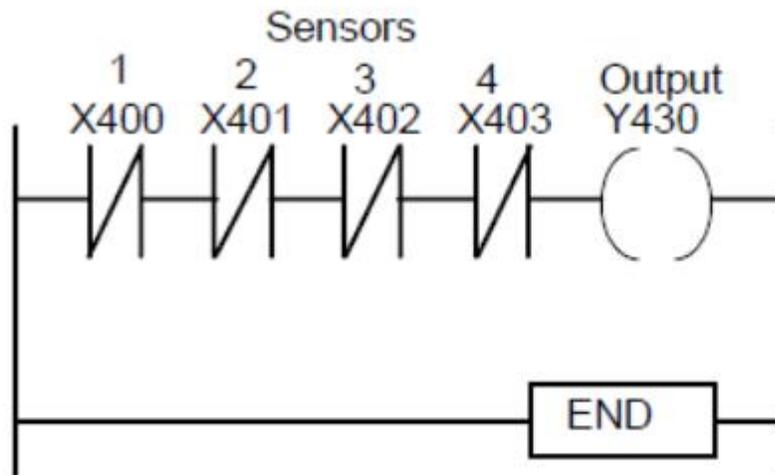


Fig : Signal lamp task



```
LD X400
OR X402
AND X401
OUT Y430
END
```

Fig : Valve operation program

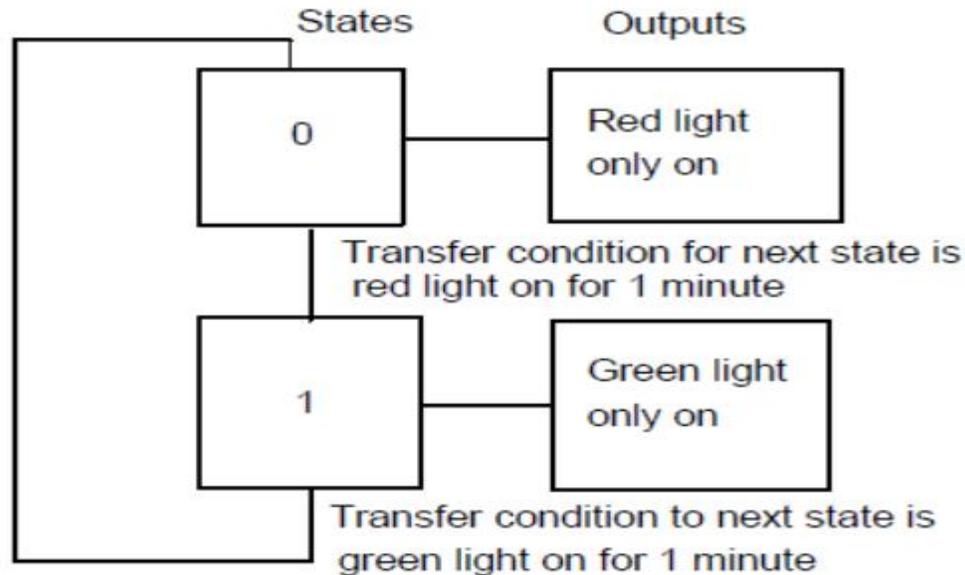


```
LDI X400
ANI X401
ANI X402
ANI X403
OUT Y430
END
```

Fig: Output switched off by any one of four sensors being activated

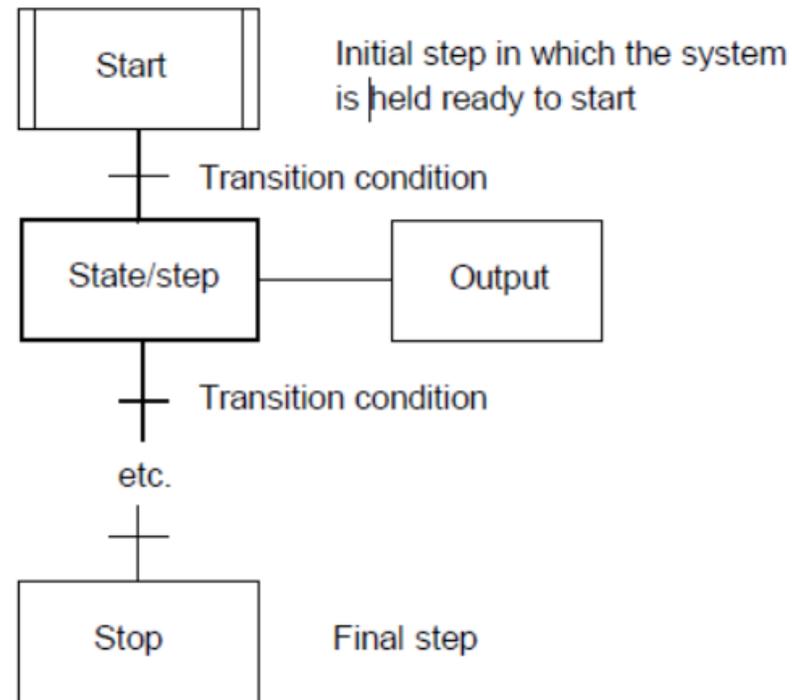
Sequential function charts

To describe a traffic lamp sequence, one way we could do this would be to represent it as a sequence of functions or states such as red light state and green light state and the inputs and outputs to each state.



The term sequential function chart (SFC) is used for a pictorial representation of a system's operation to show the sequence of the events involved in its operation. SFC charts have the following elements

1. The operation is described by a number of separate sequentially connected states or steps which are represented by rectangular boxes, each representing a particular state of the system being controlled. The initial step in a program is represented differently to the other steps,



2 .Each connecting line between states has a horizontal bar representing the transition condition that has to be realized before the system can move from one state to the next. Two steps can never be directly connected, they must always be separated by a transition. Two transitions can never directly follow from one to another, they must always be separated by a step.

3 .When the transfer conditions to the next state are realized then the next state or step in the program occurs.

4 .The process thus continues from one state to the next until the complete machine cycle is completed.

5 .Outputs/actions at any state are represented by horizontally linked boxes and occur when that state has been realized.

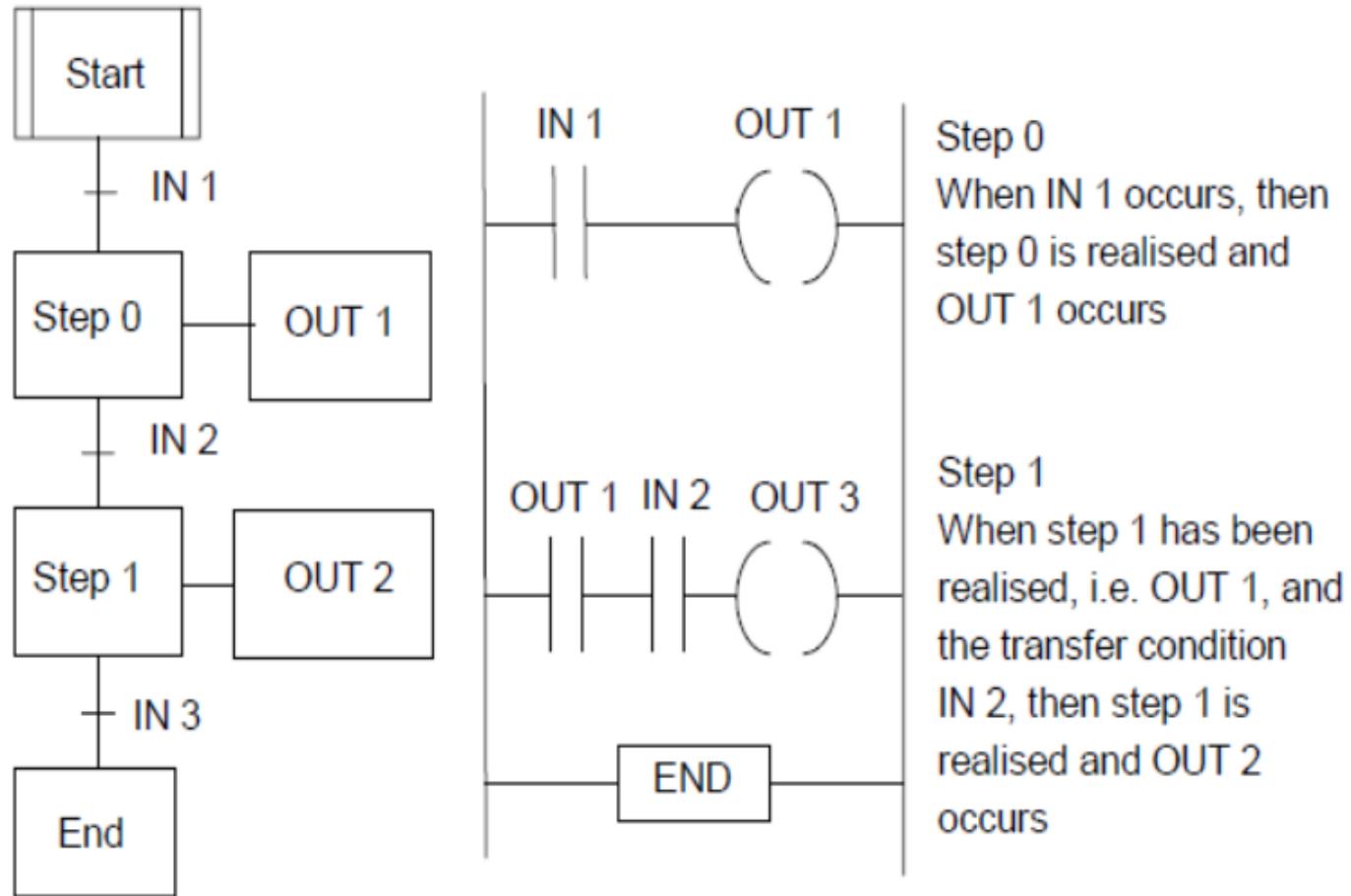


Fig : SFC and equivalent ladder diagram

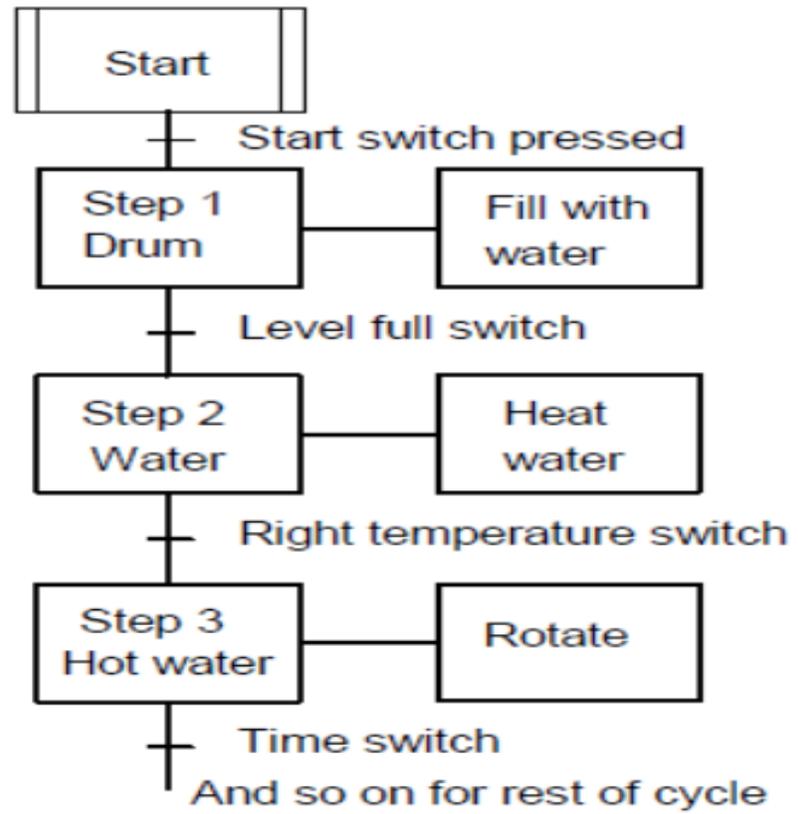


Fig : Washing machine

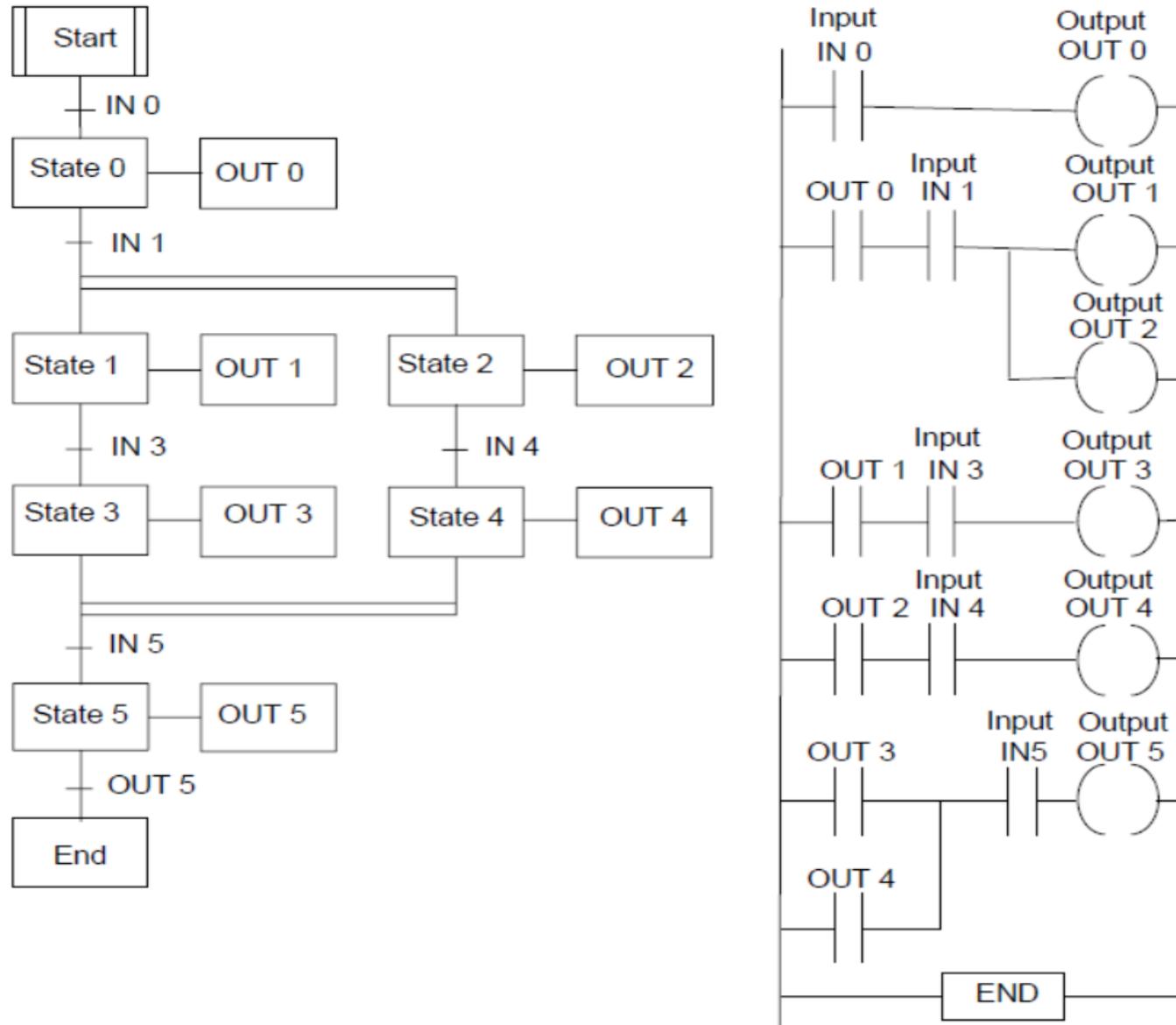


Fig : Part of an SFC and its equivalent ladder program

Structured text

Structured text is a programming language that strongly resembles the programming language PASCAL. Programs are written as a series of statements separated by semicolons. The statements use predefined statements and subroutines to change variables, these being defined values, internally stored values or inputs and outputs.

Assignment statements are used to indicate how the value of a variable is to be changed, e.g.

Light := Switch A;

is used to indicate that a light is to have its 'value' changed, i.e. switched on or off, when switch A changes its 'value', i.e. is on or off. The general format of an assignment statement is:

X := Y;

where Y represents an expression which produces a new value for the variable X.

Another example is:

Light :- Switch A OR Switch B;

to indicate that the light is switched on by either switch A or switch B. Table 6.2 shows some of the operators, like the OR in the above statement, that are used in structured text programs, and their relative precedence when an expression is being evaluated. For example:

Input A = 6

Input B = 4

Input C = 2

Output Q := Input A/3 + Input B/ (3 - Input C)

Operator	Description	Precedence
(...)	Parenthised (bracketed) expression	Highest
Function(...)	List of parameters of a function	
**	Raising to a power	
-, NOT	Negation, Boolean NOT	
*, /, MOD	Multiplication, division, modulus operation	
+, -	Addition, subtraction	
<, >, <=, >=	Less than, greater than, less than or equal to, greater than or equal to	
=, <>	Equality, inequality	
AND, &	Boolean AND	
XOR	Boolean XOR	
OR	Boolean OR	Lowest

Table : Structured text operators

Conditional statements

IF ... THEN ... ELSE is used when selected statements are to be executed when certain conditions occur. For example:

```
IF (Limit_switch1 AND Workpiece_Present) THEN
Gate1 :- Open;
Gate2 :- Close;
ELSE
Gate1 :- Close;
Gate2 :- Open;
End_IF;
```

Note that the end of the IF statement has to be indicated. Another example, using PLC addresses, is:

```
IF (I:000/00 = 1) THEN
O:001/00 :- 1;
ELSE
O:000/01 = 0;
End_IF;
```

So if there is an input to I:000/00 to make it 1 then output O:001/00 is 1, otherwise it 0.

Structured text programs

Programs have first to define the data types required to represent data e.g.

```
TYPE Motor: ( Stopped, Running);
```

```
End_TYPE;
```

```
TYPE Valve: (Open, shut);
```

```
End_TYPE;
```

```
TYPE Pressure: REAL; (*The pressure is an analogue value*)
```

```
End_TYPE;
```

the variables, i.e. signals from sensors and output signals to be used in a program, e.g.

```
VAR_IN (*Inputs*)
```

```
Pump Fault : BOOL; (*Pump operating fault is a Boolean variable*)
```

```
End_VAR;
```

```
VAR_OUT (*Outputs*)
```

```
Motor_speed : REAL;
```

```
End_VAR;
```

```
VAR_IN
```

```
Value: INT; (*The value is an integer*)
```

```
End_VAR;
```

```
VAR
```

```
Input1 AT %IX100; (*Input1 is located at input memory bit 100*)
```

```
End_VAR;
```

and any initial values to be given to variables, e.g.

```
VAR
```

```
Temp : REAL :-100; (*Initial value is a an analogue number 100*)
```

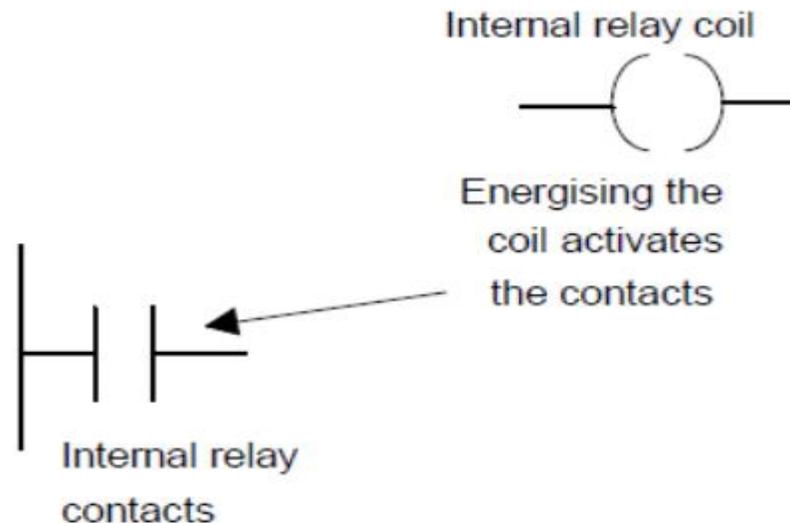
```
End_VAR;
```

before getting down to the instruction statements.

Internal relays

Internal relays are coils and contacts that are simulated by the PLC in memory. Like external relays, they consist of an output coil and a set of contacts that can be used as the input to other objects on a ladder rung.

In PLCs there are elements that are used to hold data, i.e. bits, and behave like relays, being able to be switched on or off and switch other devices on or off. Hence the term internal relay. Such internal relays do not exist as real-world switching devices but are merely bits in the storage memory that behave in the same way as relays. For programming, they can be treated in the same way as an external relay output and input.



Different ways of expressing internal relay :

To distinguish internal relay outputs from external relay outputs, they are given different types of addresses. Different manufacturers tend to use different terms for internal relays and different ways of expressing their addresses.

Mitsubishi uses the term auxiliary relay or marker and the notation M100, M101, etc.

Siemens uses the term flag and notation F0.0, F0.1, etc.

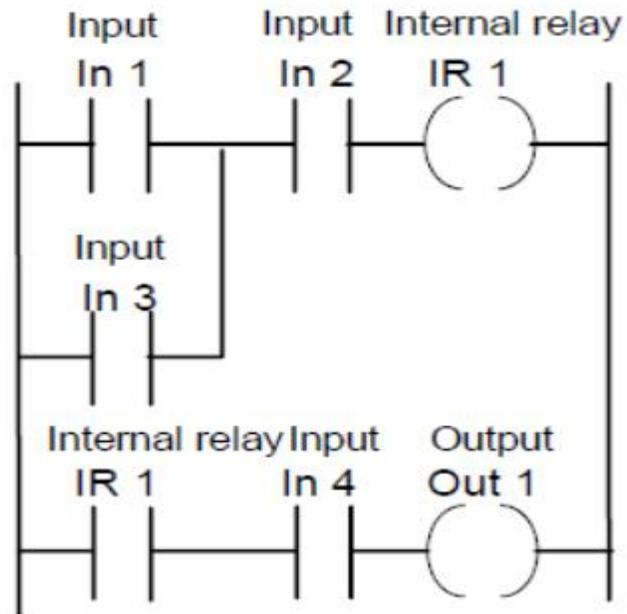
Telemecanique uses the term bit and notation B0, B1, etc.

Toshiba uses the term internal relay and notation R000, R001, etc.

Allen-Bradley uses the term bit storage and notation in the PLC-5 of the form B3/001, B3/002, etc.

Ladder programs

With ladder programs, an internal relay output is represented using the symbol for an output device, namely (), with an address which indicates that it is an internal relay rather than an external relay. The internal relay switching contacts are designated with the symbol for an input device, namely | |, and given the same address as the internal relay output, e.g. M100.



Internal relay

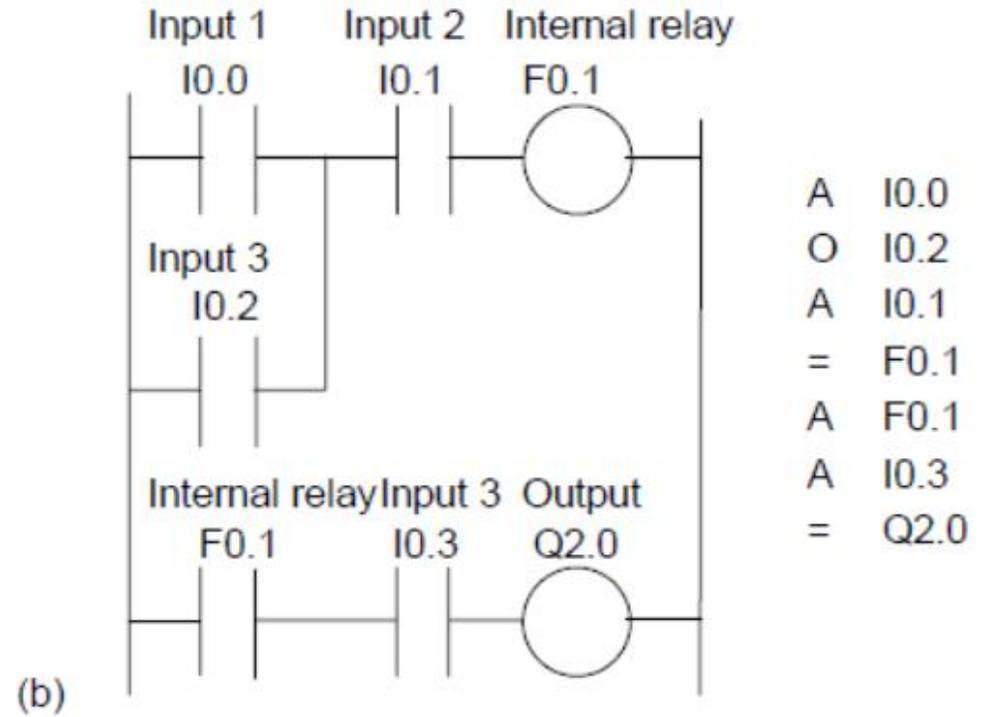
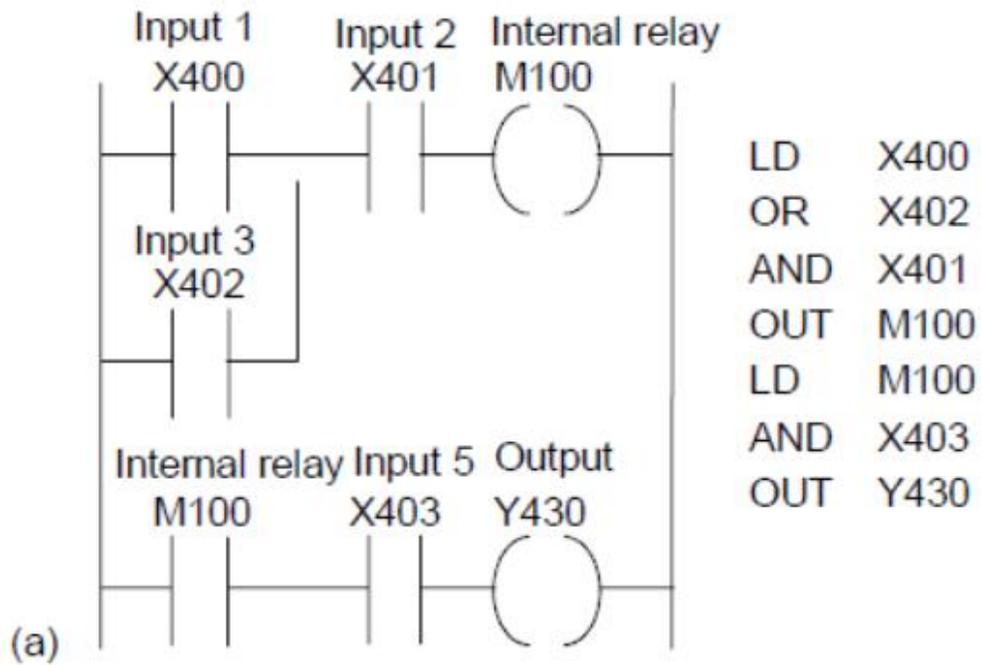


Figure (a) Mitsubishi notation, (b) Siemens notation

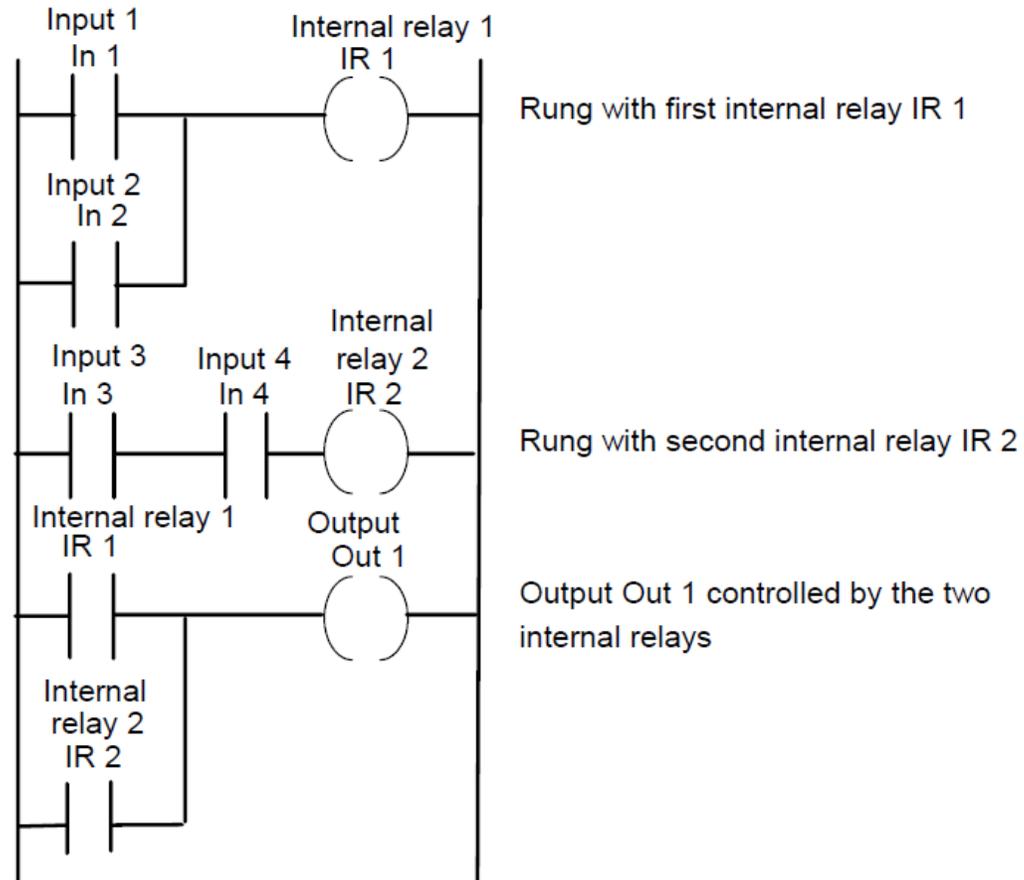
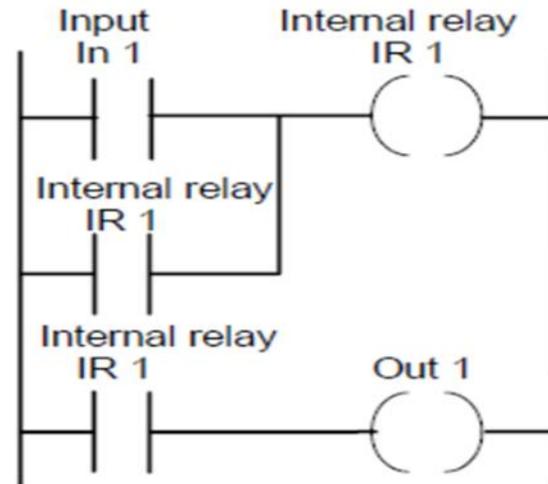


Fig: Use of two internal relays

Battery-backed relays

If the power supply is cut off from a PLC while it is being used, all the output relays and internal relays will be turned off. Thus when the power is restored, all the contacts associated with those relays will be set differently from when the power was on. Thus, if the PLC was in the middle of some sequence of control actions, it would resume at a different point in the sequence. To overcome this problem, some internal relays have battery back-up so that they can be used in circuits to **ensure a safe shutdown** of plant in the event of a power failure and so **enable it to restart in an appropriate manner**. Such battery-backed relays retain their state of activation, even when the power supply is off. The relay is said to have been made retentive.

A ladder diagram for a system designed to cope with a power failure. IR 1 is a battery-backed internal relay. When input 1 contacts close, output IR 1 is energized. This closes the IR 1 contacts, latching so that IR 1 remains on even if input 1 opens. The result is an output from Out 1. If there is a power failure, IR 1 still remains energized and so the IR 1 contacts remain closed and there is an output from Out 1.



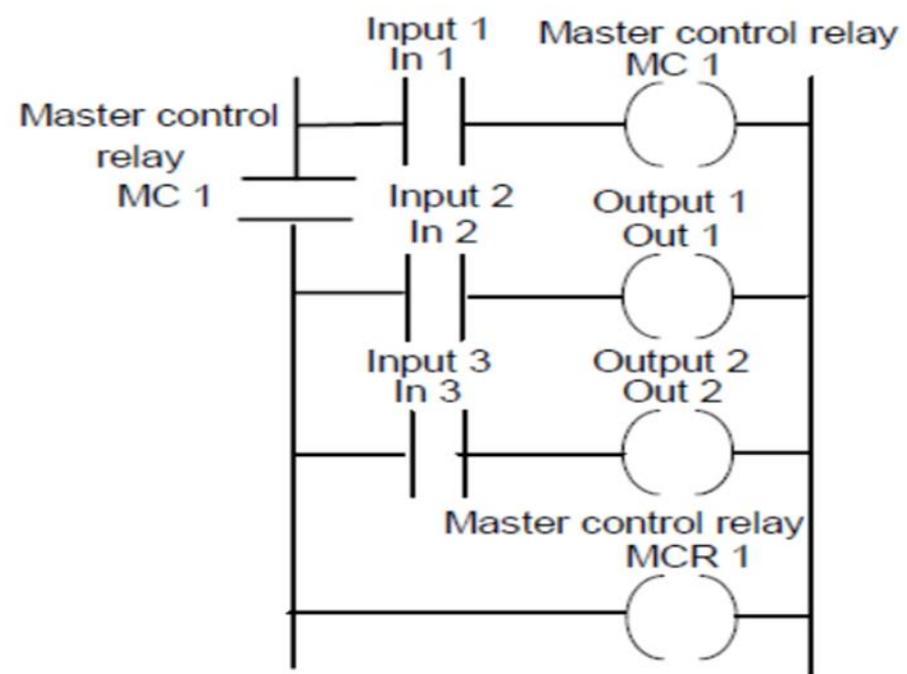
Battery-backed relay program

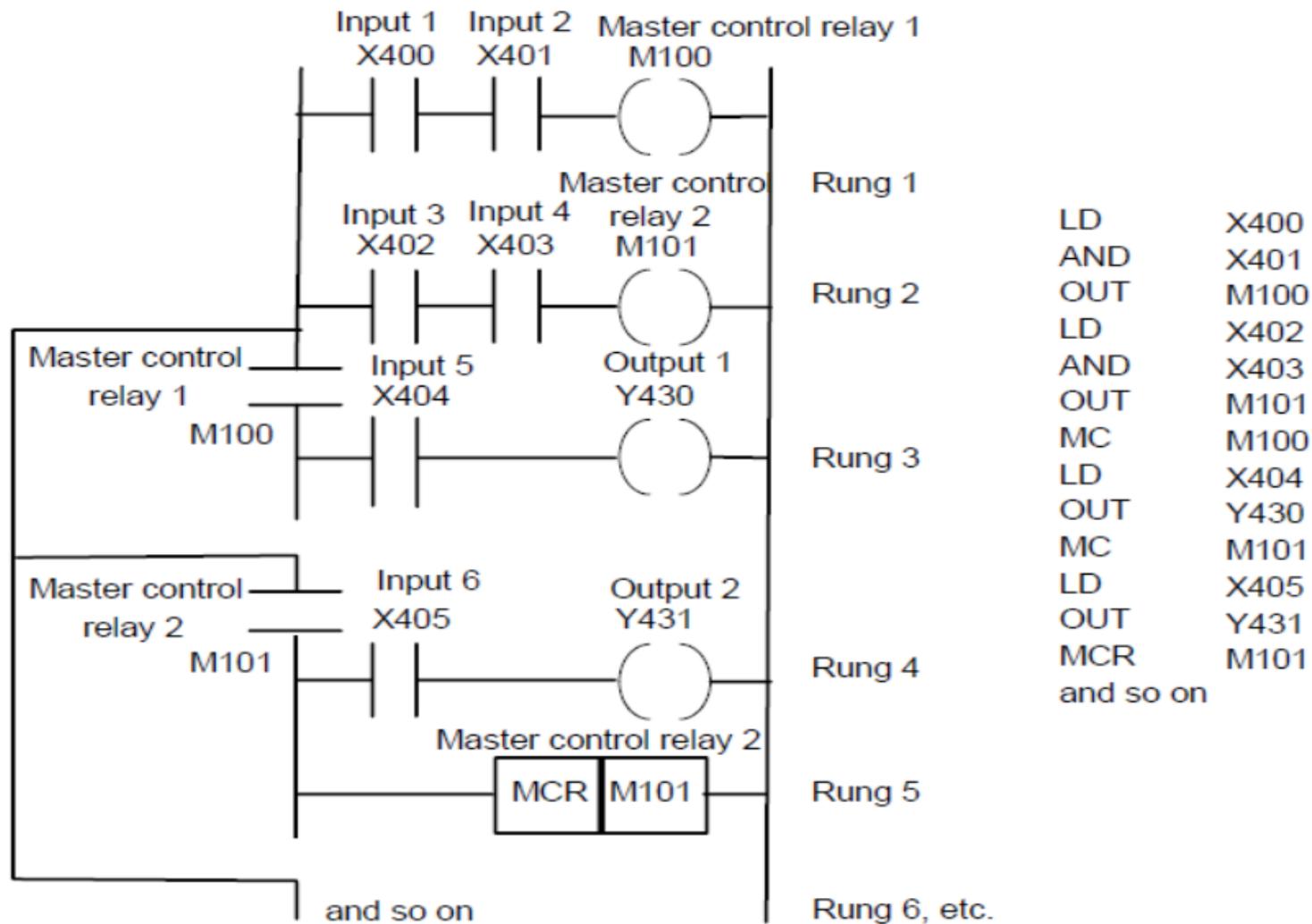
Master control relay

When large numbers of outputs have to be controlled, it is sometimes necessary for whole sections of ladder diagrams to be turned on or off when certain criteria are realized. This could be achieved by including the contacts of the same internal relay in each of the rungs so that its operation affects all of them. An alternative is to use a master control relay.

Fig : Principle of use of a master control relay

With no input to input 1, the output internal relay MC 1 is not energized and so its contacts are open. This means that all the rungs between where it is designated to operate and the rung on which its reset MCR or another master control relay is located are switched off. Assuming it is designated to operate from its own rung, then we can imagine it to be located in the power line in the position shown and so rungs 2 and 3 are off. When input 1 contacts close, the master relay MC 1 is energized. When this happens, all the rungs between it and the rung with its reset MCR 1 are switched on. **Thus outputs 1 and 2 cannot be switched on by inputs 2 and 3 until the master control relay has been switched on.** The master control relay 1 acts only over the region between the rung it is designated to operate from and the rung on which MCR 1 is located.





Example showing more than one master control relay

Jump and call

A function often provided with PLCs is the conditional jump. We can describe this as:

IF (some condition occurs) THEN

perform some instructions

ELSE

perform some other instructions.

Such a facility enables programs to be designed such that if certain conditions are met then certain events occur, if they are not met then other events occur. Thus, for example, we might need to design a system so that if the temperature is above 60oC a fan is switched on, and if below that temperature no action occurs.

Thus, if the appropriate conditions are met, this function enables part of a ladder program to be jumped over.

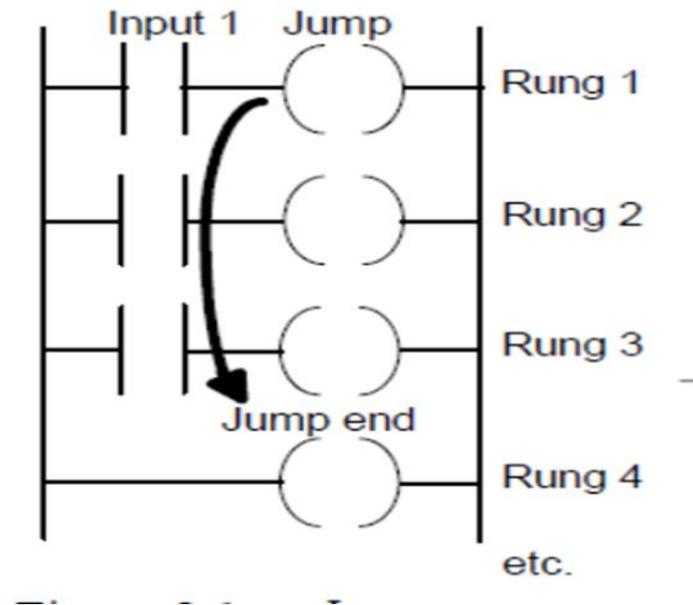


Fig: Jump

When there is an input to In 1, its contacts close and there is an output to the jump relay. This then results in the program jumping to the rung in which the jump end occurs, so missing out intermediate program rungs. Thus, in this case, when there is an input to Input 1, the program jumps to rung 4 and then proceeds with rungs 5, 6, etc. When there is no input to Input 1, the jump relay is not energized and the program then proceeds to rungs 2, 3, etc.

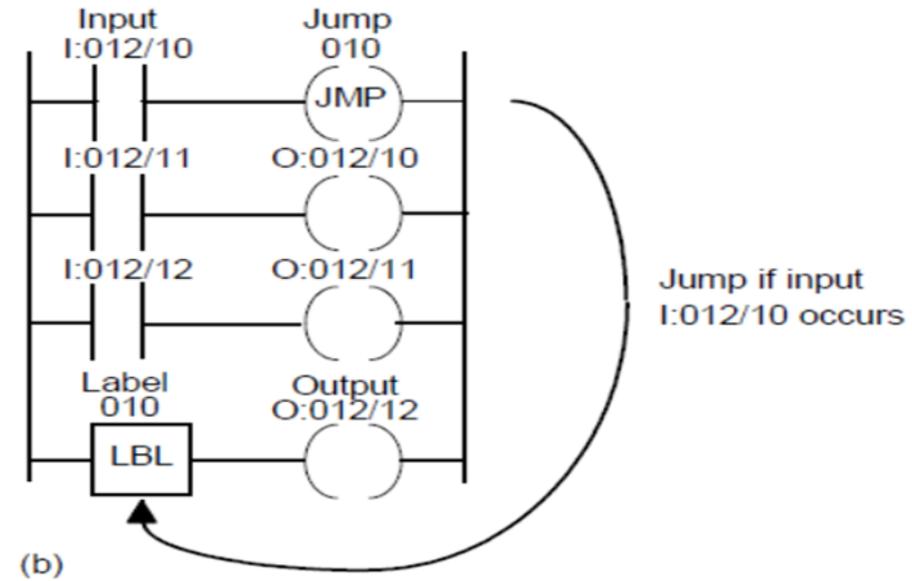
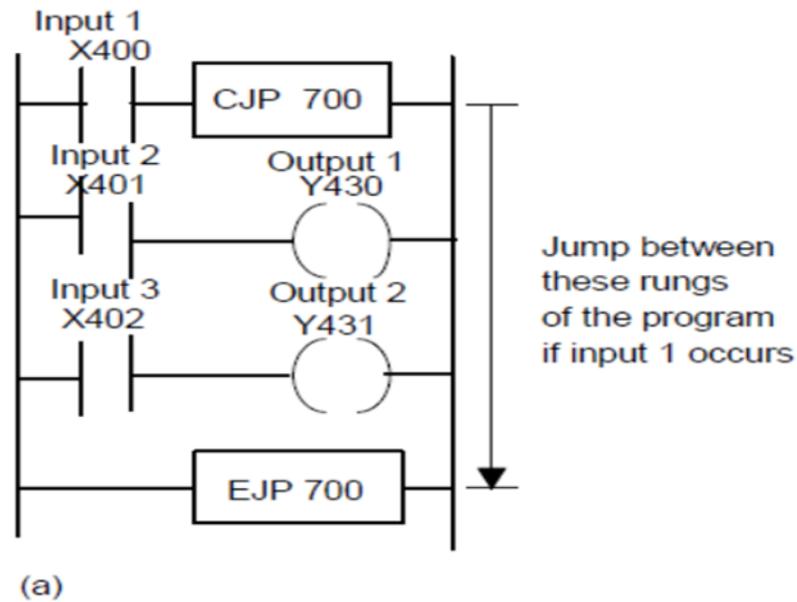


Fig: Jump: (a) Mitsubishi program, (b) Allen-Bradley program

shows the above ladder program in the form used by Mitsubishi. The jump instruction is denoted by CJP (conditional jump) and the place to which the jump occurs is denoted by EJP (end of jump). The condition that the jump will occur is then that there is an input to X400. When that happens the rungs involving inputs X401 and X403 are ignored and the program jumps to continue with the rungs following the end jump instruction with the same number as the start jump instruction, i.e. in this case EJP 700.

With the Allen-Bradley PLC-5 format the jump takes place from the jump instruction (JMP) to the label instruction (LBL). The JMP instruction is given a three-digit number from 000 to 255 and the LBL instruction the same number. Figure shows a ladder program in this format

Jumps within jumps

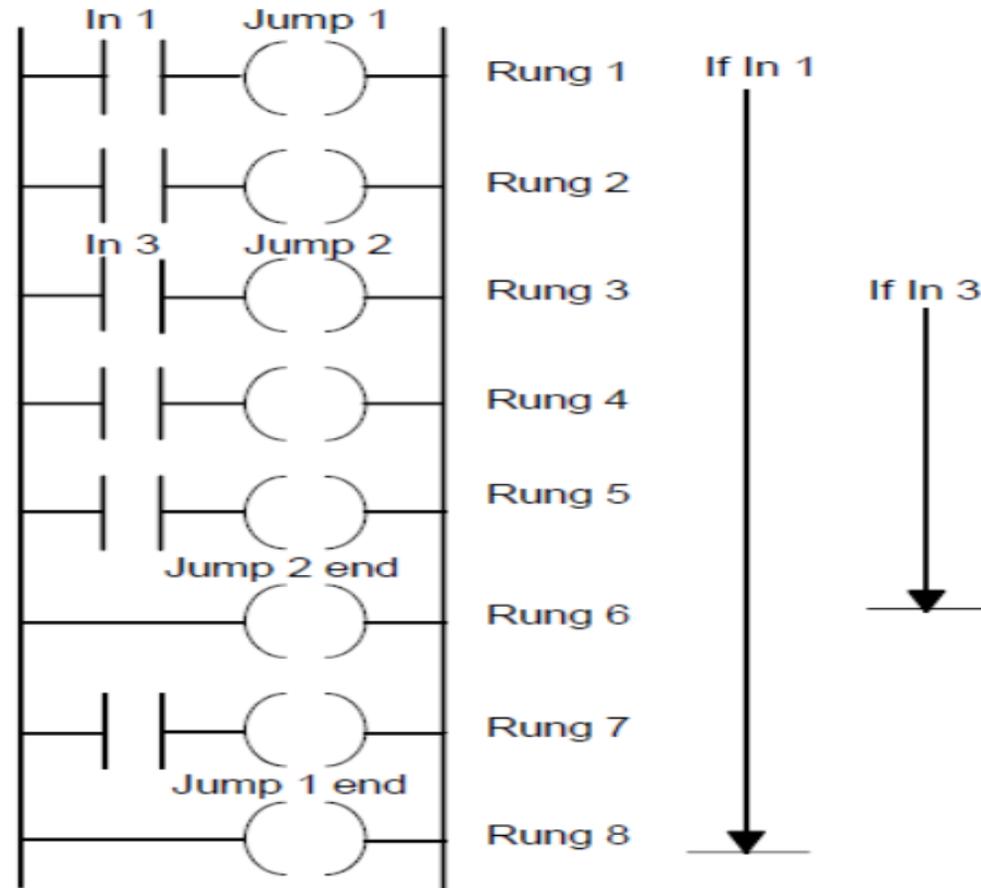


Fig : Jumps within jumps

Jumps within jumps are possible. If the condition for the **jump instruction 1 is realized** then the program **jumps to rung 8**. If the condition is not met then the program continues to rung 3. If the condition for the jump instruction 2 is realized then the program jumps to rung 6. If the condition is not met then the program continues through the rungs.

Thus if we have an input to In 1, the rung sequence is rung 1, 8, etc. If we have no input to In 1 but an input to In 3, then the rung sequence is 1, 2, 6, 7, 8, etc. If we have no input to In 1 and no input to In 3, the rung sequence is 1, 2, 3, 4, 5, 6, 7, 8, etc. The jump instruction enables different groups of program rungs to be selected, depending on the conditions occurring.

Timers

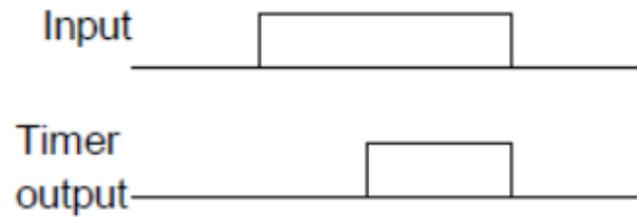
In many control tasks there is a need **to control time**. For example, a motor or a pump might need to be **controlled to operate** for a particular interval of time, or perhaps be switched on after some time interval. PLCs thus have timers as built-in devices. Timers count fractions of seconds or seconds using the internal CPU clock.

PLC manufacturers differ on **how** timers should be **programmed** and hence **how** they can be **considered**. A common approach is to consider timers to behave like relays with coils which when energized result in the closure or opening of contacts after some preset time.

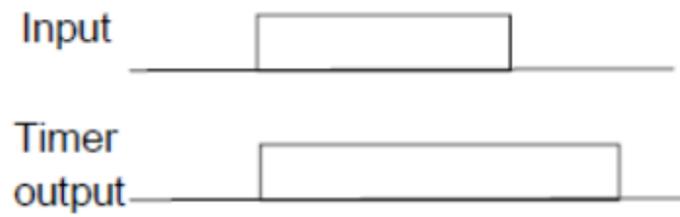
Types of timers

There are a number of different forms of timers that can be found with PLCs. With small PLCs there is likely to be just one form,

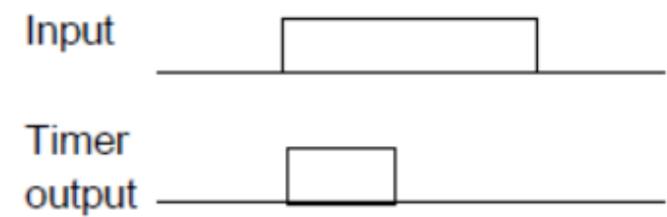
1. On-delay timers. These are timers which come on after a particular time delay.
2. Off-delay timers. are on for a fixed period of time before turning off .
3. Pulse timer. This timer switches on or off for a fixed period of time.



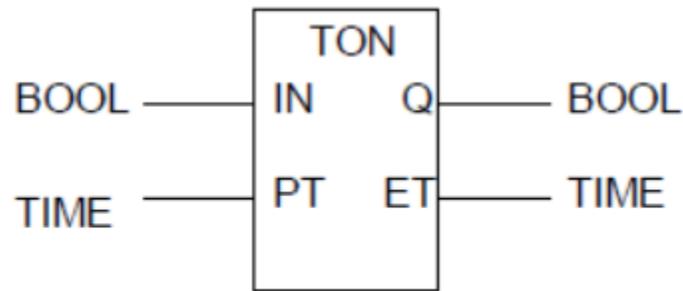
(a) On-delay timer



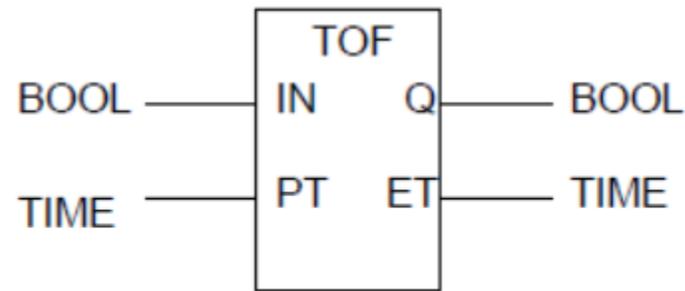
(a) Off-delay timer



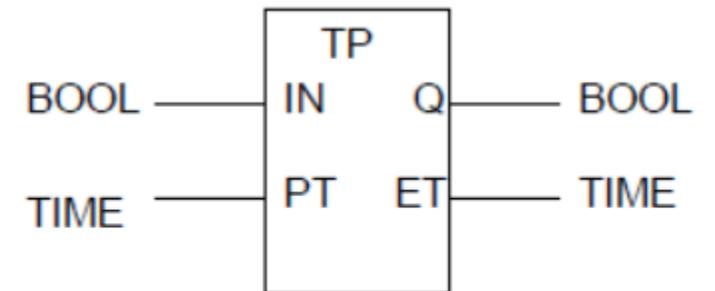
(c) Pulse timer



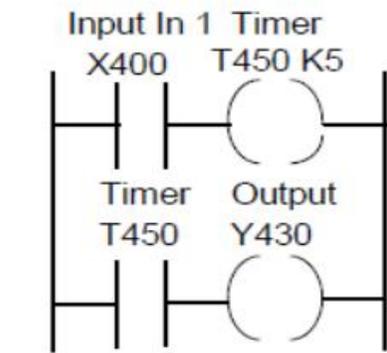
On-delay timer



Off-delay timer

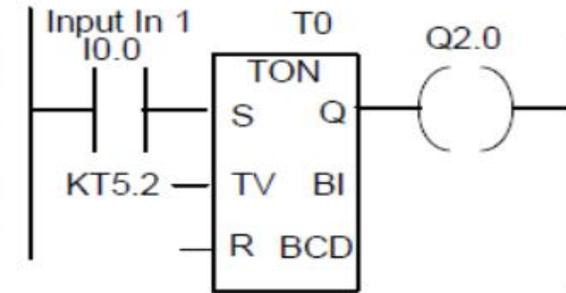


Pulse timer



(a)

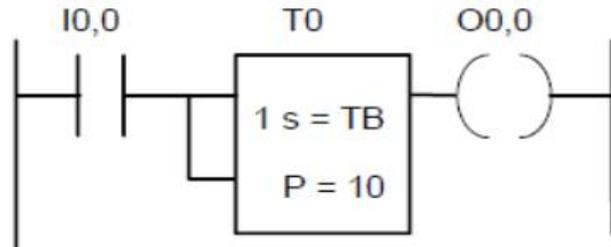
LD X400
 OUT T450
 K 5
 LD T450
 OUT Y430



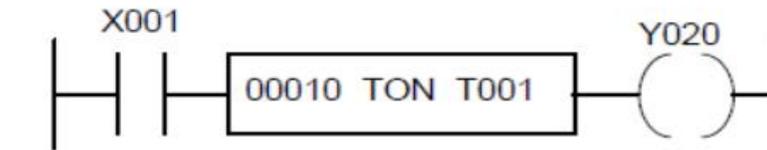
(b)

A I0.0
 LKT 5.2
 SR T0
 A T0
 = Q2.0

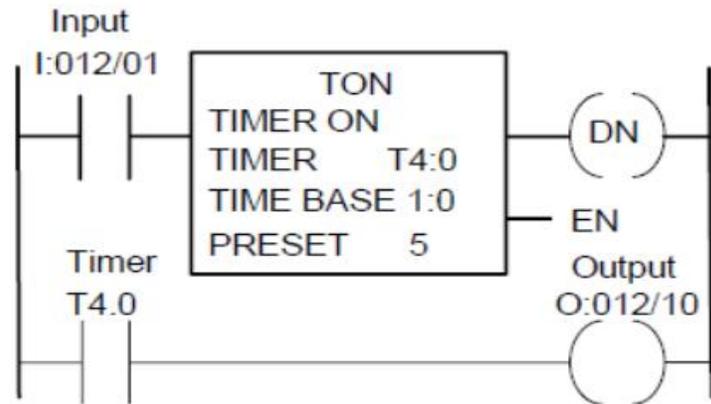
S is Boolean start input.
 TV is duration of time specification.
 R is Boolean reset.
 BI is current time value in binary word.
 BCD is current time value in BCD word
 Q is Boolean output, indicating state of timer.



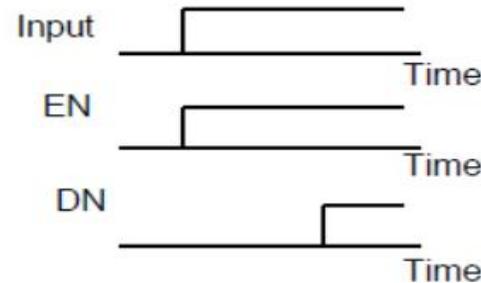
(c)



(d)



(e)



The enable bit EN is set to 1 when there is a logic path to the time. The done bit DN indicates the status of the timer and is set to 1 when the accumulated value equals the preset value.

Timers: (a) Mitsubishi, (b) Siemens, (c) Telemecanique, (d) Toshiba, (e) Allen-Bradley

How to start a motor using Timer?

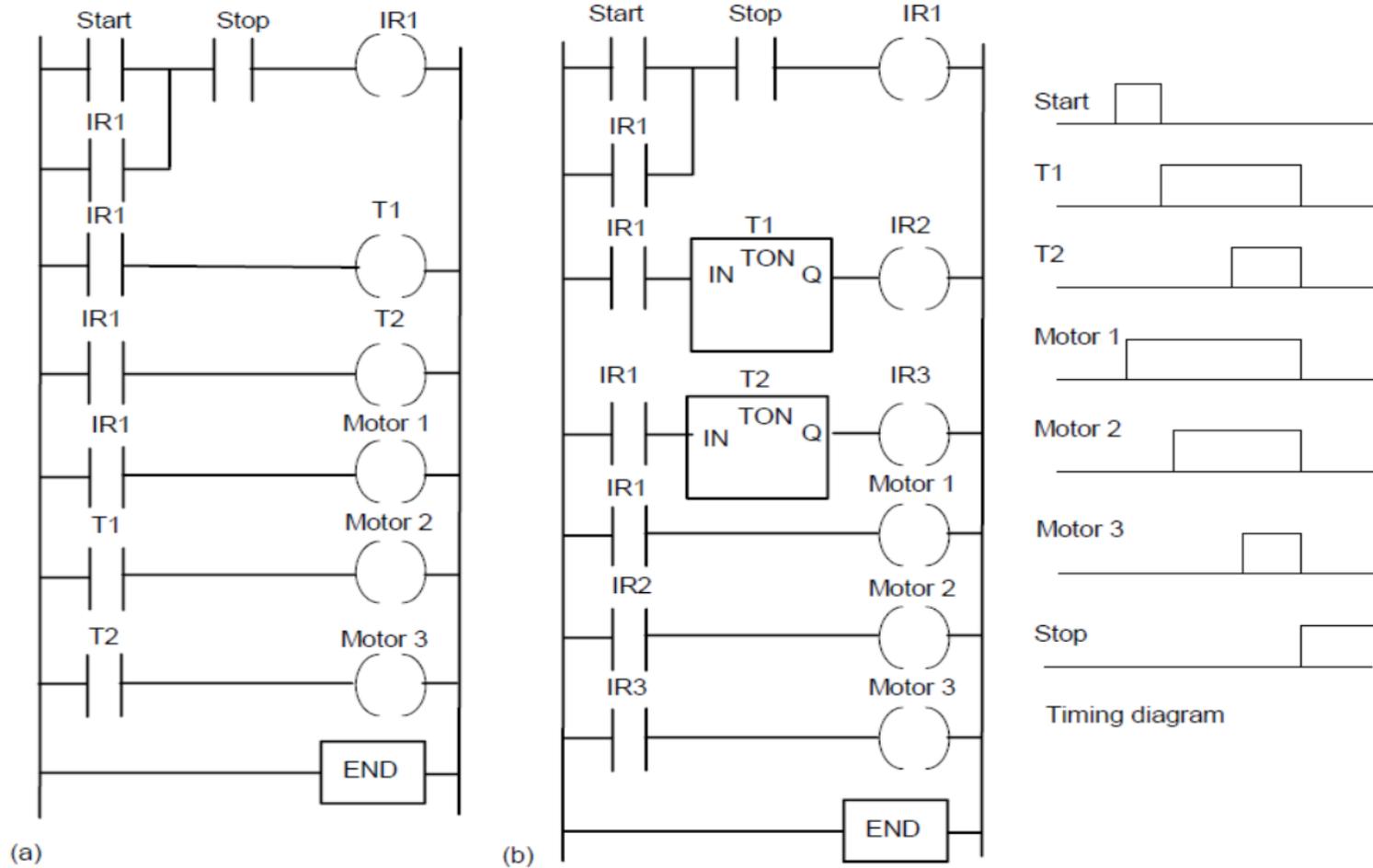


Fig : Motor sequence

Shows two versions of how timers can be used to start three outputs, e.g. three motors, in sequence following a single start button being pressed. In (a) the timers are programmed as coils, whereas in (b) they are programmed as delays. When the start push button is pressed there is an output from internal relay IR1. This latches the start input. It also starts both the timers, T1 and T2, and motor 1. When the preset time for timer 1 has elapsed then its contacts close and motor 2 starts. When the preset time for timer 2 has elapsed then its contacts close and motor 3 starts. The three motors are all stopped by pressing the stop push button. Since this is seen as a complete program, the end instruction has been used.

Programming involving timers consider the sequencing of traffic lights

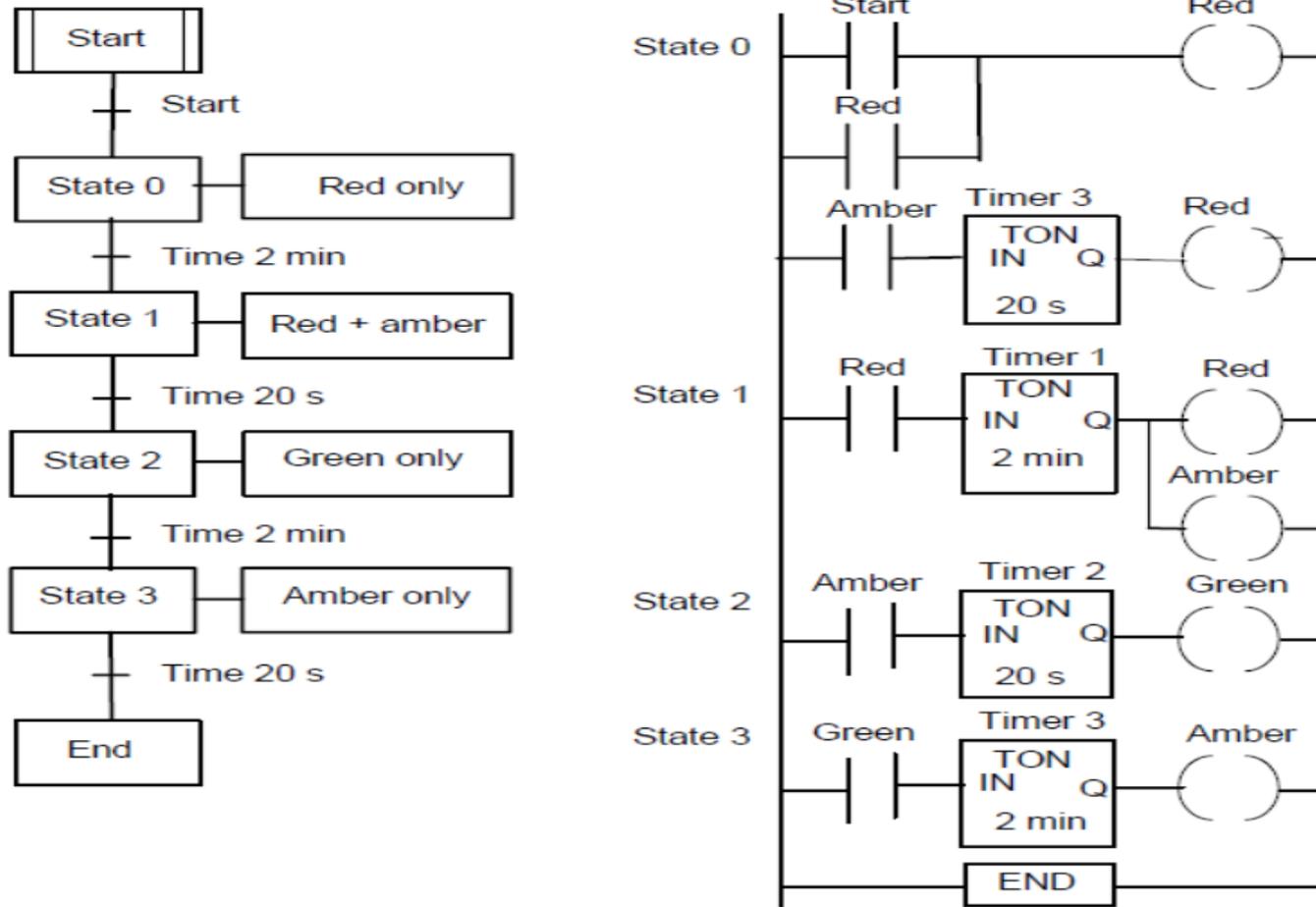


Fig: Traffic light sequence

Counters

Counters are provided as built-in elements in PLCs and allow the number of occurrences of input signals to be counted. This might be where items have to be counted as they pass along a conveyor belt, or the number of revolutions of a shaft, or perhaps the number of people passing through a door. This chapter describes how such counters can be programmed.

Forms of counter

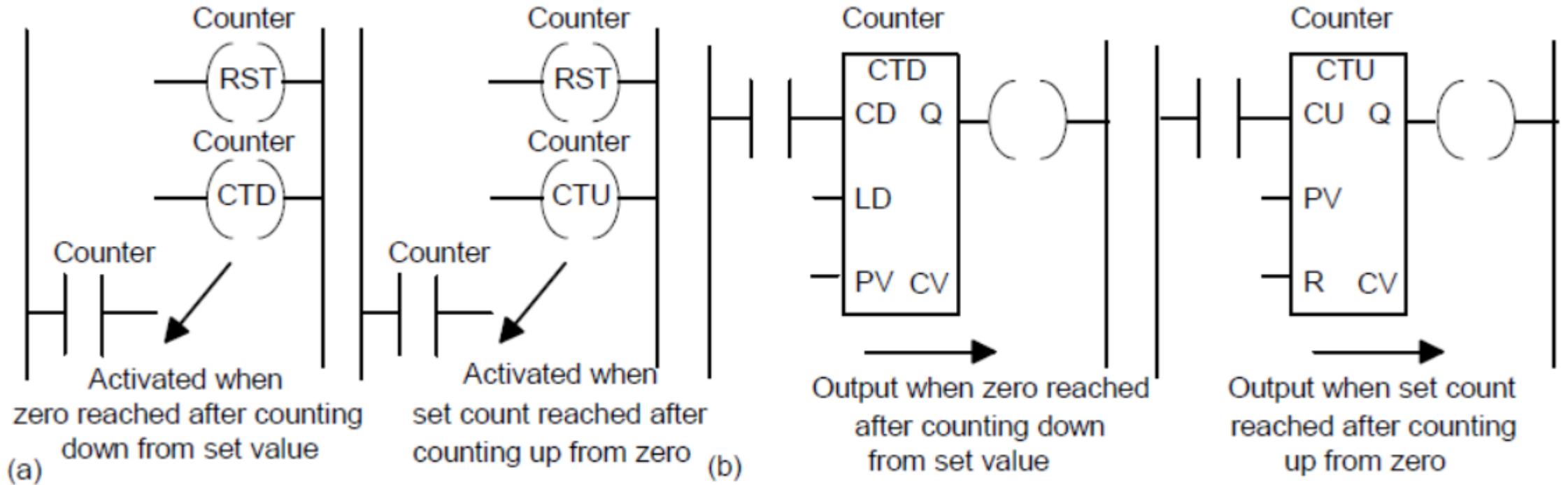
A counter is set to some preset number value and, when this value of input pulses has been received, it will operate its contacts. Thus normally open contacts would be closed, normally closed contacts opened.

There are two types of counter, though PLCs may not include both types.

These are 1. down-counters and 2. up-counters.

Down-counters: count down from the preset value to zero, i.e. events are subtracted from the set value. When the counter reaches the zero value, its contacts change state. Most PLCs offer down counting.

Up-counters : count from zero up to the preset value, i.e. events are added until the number reaches the preset value. When the counter reaches the set value, its contacts change state.

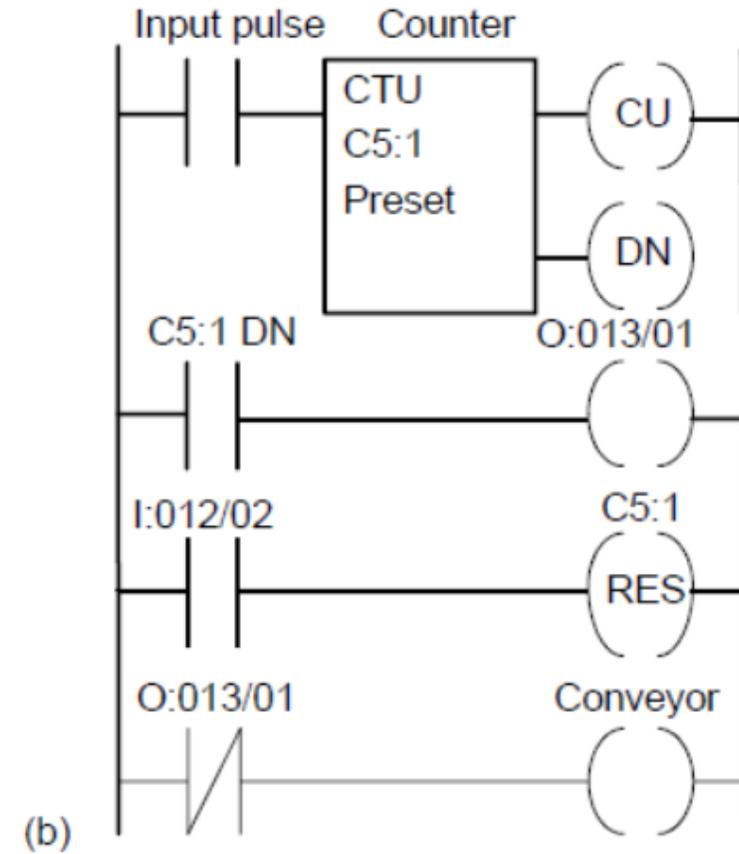
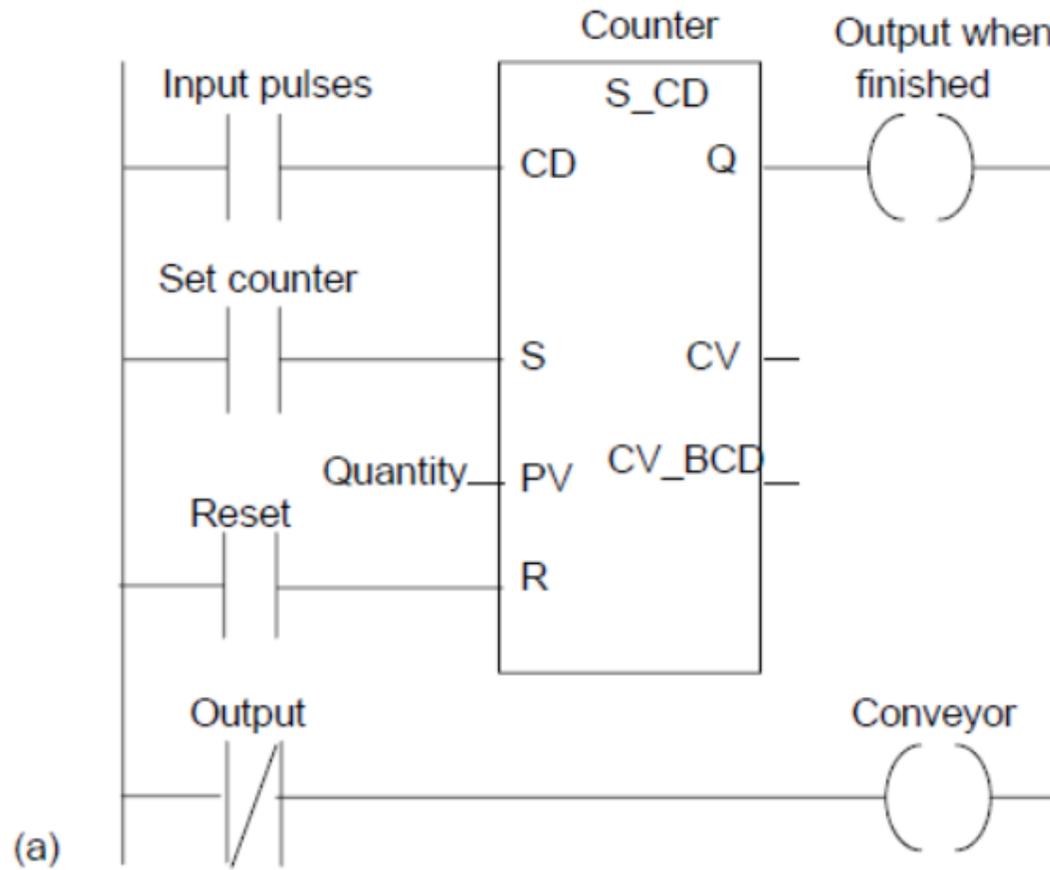


Different PLC manufacturers deal with counters in slightly different ways. Some count down (CTD), or up (CTU), and reset and treat the counter as though it is a relay coil and so a rung output. In this way, counters can be considered to consist of two basic elements: one relay coil to count input pulses and one to reset the counter, the associated contacts of the counter being used in other rungs.

Counter application

As an illustration of the use that can be made of a counter, consider the problem of items passing along a conveyor belt, the passage of an item past a particular point being registered by a light beam to a photoelectric cell being interrupted, and after a set number there is to be a signal sent informing that the set count has been reached and the conveyor stopped.

Carrying item calculation by conveyor belt



The basic control loop can be simplified for a single-input-single-output (SISO) system as in Fig. Here we are neglecting any disturbance present in the system.

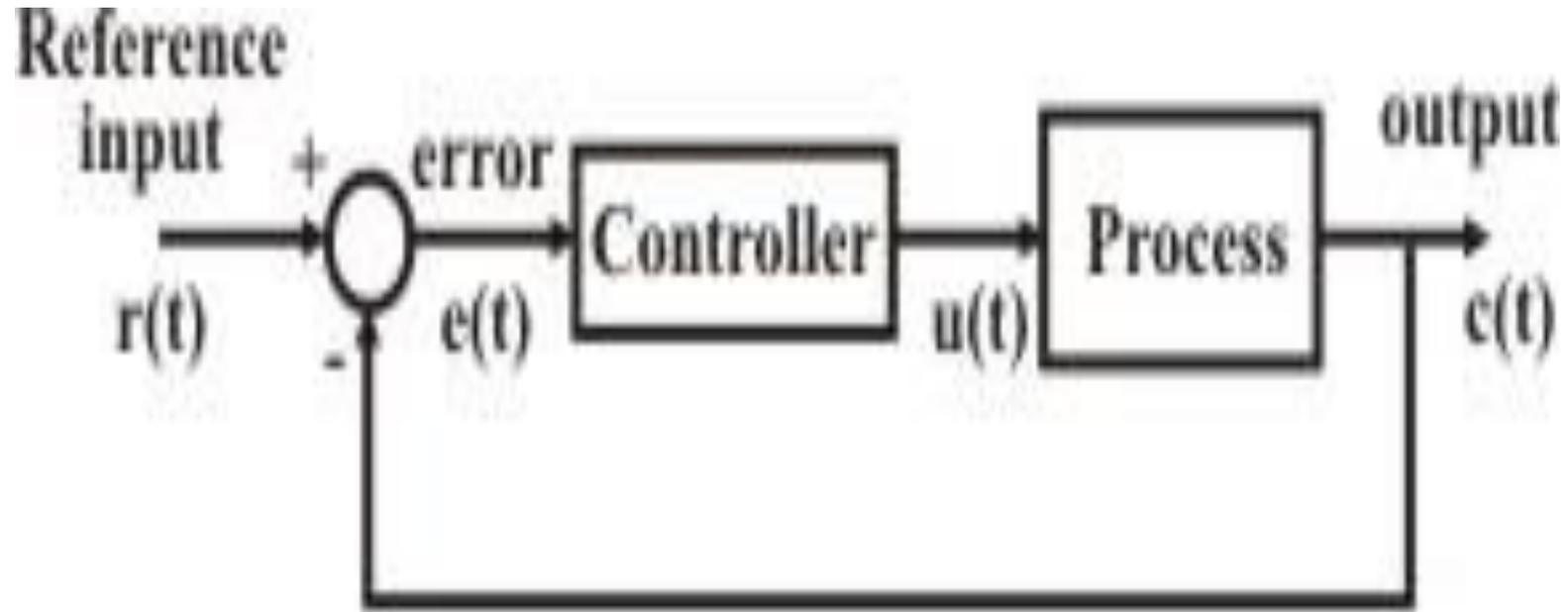


Fig. 1 A closed loop SISO system

The controller may have different structures. Different design methodologies are there for designing the controller in order to achieve desired performance level.

controllers use a 3 basic behavior types or modes:

P - proportional,

I - integrative

and D - derivative.

behavior of controllers main types or modes:

1.P Controller

2.P-I Controller

3. P-D Controller

4. P-I-D Controller

P controller can stabilize only 1st order unstable process.

PI controller can be used to avoid large disturbances and noise presents during operation process.

Whereas PID controller can be used when dealing with higher order capacitive processes.

The comparative study of P, PI and PID Controller is carried out, in which PID controller gives good response than any other controller.

1.P Controller

P Controller: A P controller system is a type of linear feedback control system. The P controller system is more complex than on-off control systems like a bi-metallic domestic thermostat, but simpler than a PID control system used in something like an automobile cruise control. In general it can be said that P controller cannot stabilize higher order processes.

For the 1st order processes, meaning the processes with one energy storage, a large increase in gain can be tolerated.

Proportional controller can stabilize only 1st order unstable process. Changing controller gain K can change closed loop dynamics. A large controller gain will result in control system with:

- a) Smaller steady state error, i.e. better reference following
- b) Faster dynamics, i.e. broader signal frequency band of the closed loop system and larger sensitivity with respect to measuring noise
- c) Smaller amplitude and phase margin

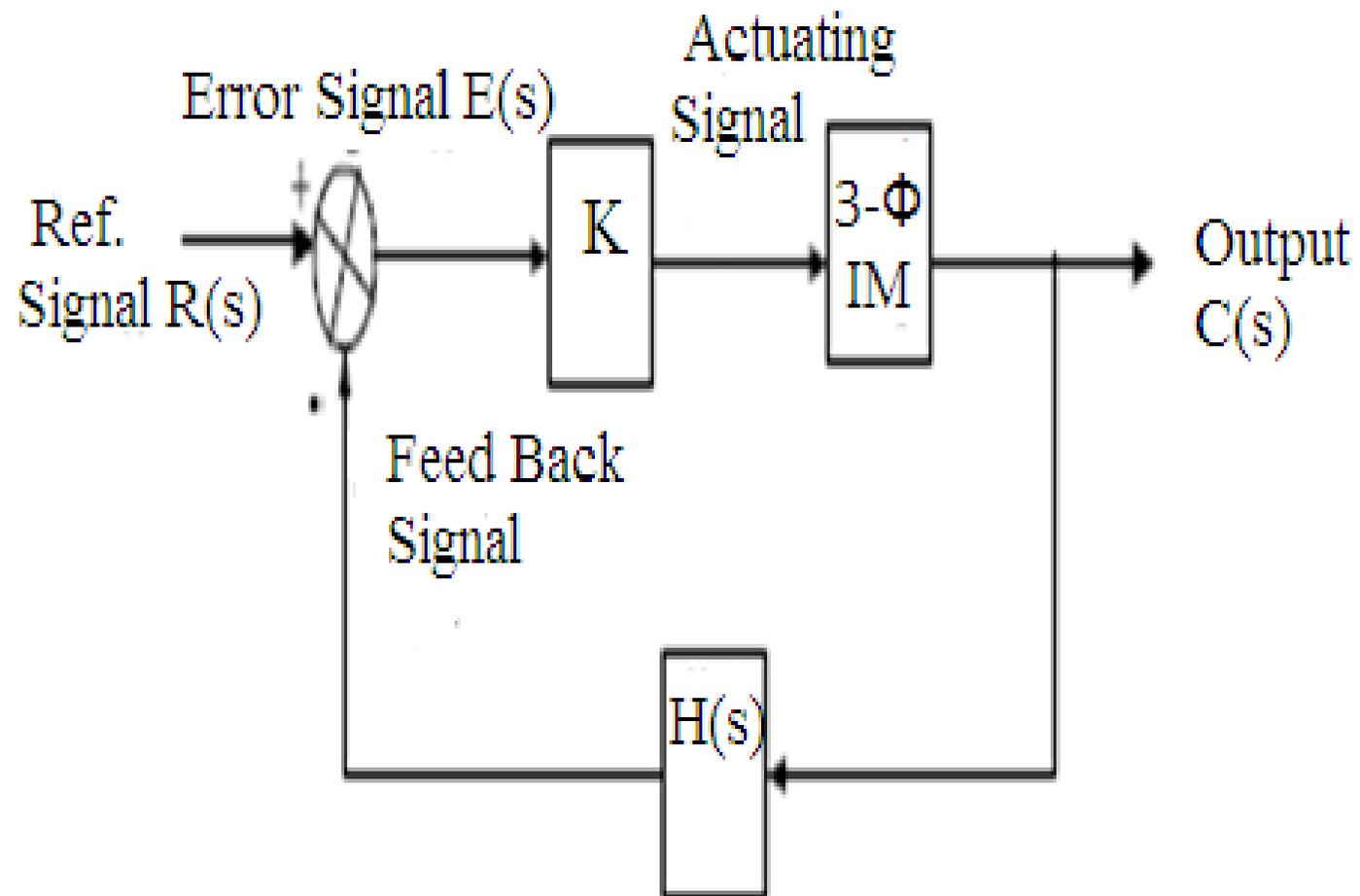


Fig : Block diagram of P controller

In the P controller algorithm, the controller output is proportional to the error signal, which is the difference between the set point and the process variable.

In P controller the actuating signal for the control action in a control system is proportional to the error signal. The error signal being the difference between the reference input signal and feedback signal obtained from the output.

P-I Controller:

P-I controller is mainly used to eliminate the steady state error resulting from P controller. However, in terms of the speed of the response and overall stability of the system, it has a negative impact. This controller is mostly used in areas where speed of the system is not an issue. Since P-I controller has no ability to predict the future errors of the system it cannot decrease the rise time and eliminate the oscillations. If applied, any amount of I guarantees set point overshoot.

Proportional Integral (PI) Controller

At present, the PI controller is most widely adopted in industrial application due to its simple structure, easy to design and low cost.

Despite these advantages, the PI controller fails when the controlled object is highly nonlinear and uncertain.

PI controller will eliminate forced oscillations and steady state error resulting in operation of on-off controller and P controller respectively.

However, introducing integral mode has a negative effect on speed of the response and overall stability of the system. Thus, PI controller will not increase the speed of response.

It can be expected since PI controller does not have means to predict what will happen with the error in near future. This problem can be solved by introducing derivative mode which has ability to predict what will happen with the error in near future and thus to decrease a reaction time of the controller.

PI controllers are very often used in industry, especially when speed of the response is not an issue. A control without D mode is used when

1. Fast response of the system is not required
2. Large disturbances and noise are present during operation of the process
3. There is only one energy storage in process (capacitive or inductive)
4. There are large transport delays in the system.

Therefore, we would like to keep the advantages of the PI controller. This leads to propose a PI controller shown in Fig.. This controller uses of the proportional term while the integral term is kept, unchanged.

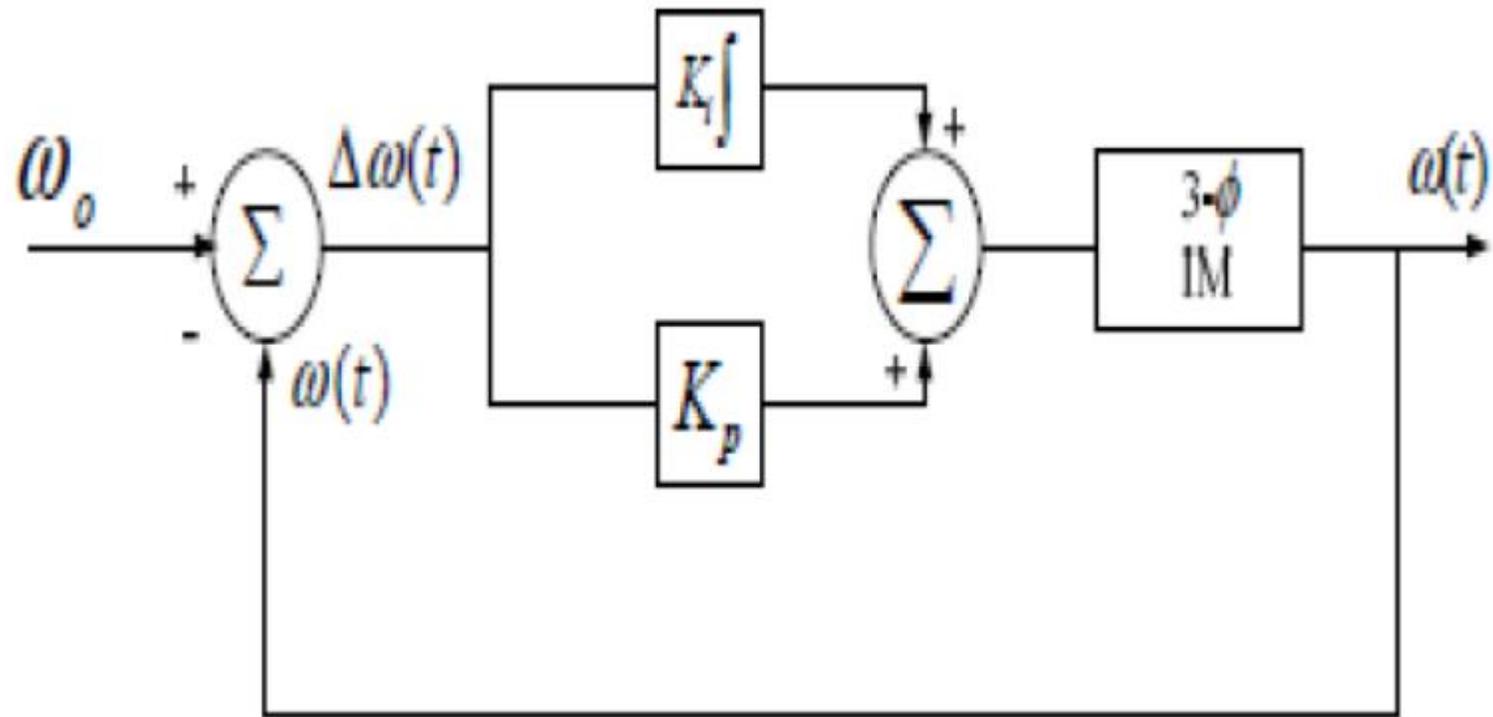


Fig : Block diagram of PI controller

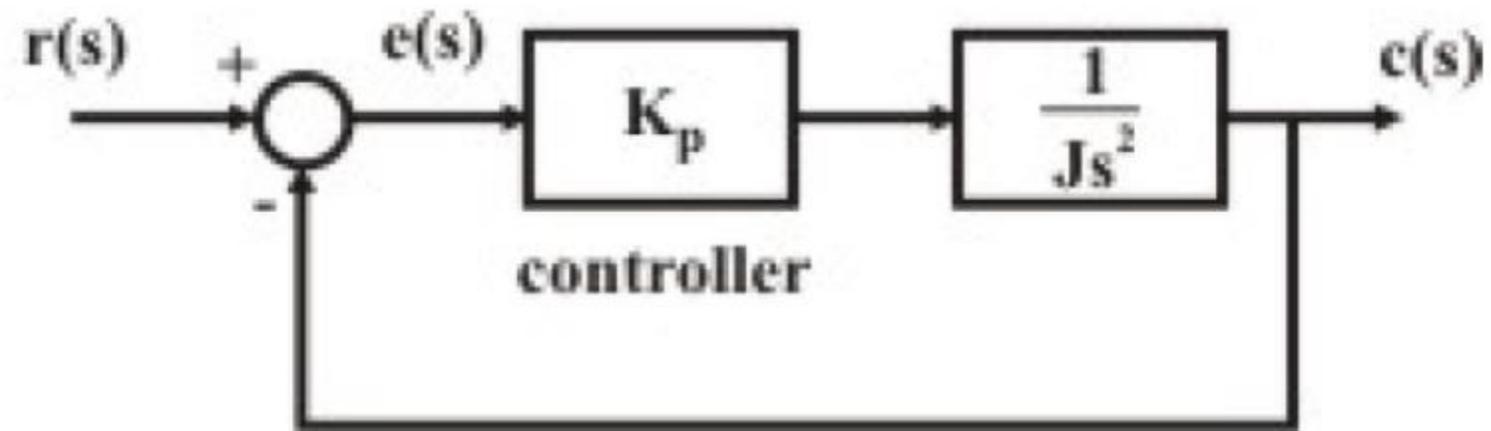
P-I controller is mainly used to eliminate the steady state error resulting from P controller. However, in terms of the speed of the response and overall stability of the system, it has a negative impact. This controller is mostly used in areas where speed of the system is not an issue. Since P-I controller has no ability to predict the future errors of the system it cannot decrease the rise time and eliminate the oscillations. If applied, any amount of I guarantees set point overshoot.

PD Controller:

D mode is used when prediction of the error can improve control or when it necessary to stabilize the system. From the frequency characteristic of D element it can be seen that it has phase lead of 90° .

Often derivative is not taken from the error signal but from the system output variable. This is done to avoid effects of the sudden change of the reference input that will cause sudden change in the value of error signal. Sudden change in error signal will cause sudden change in control output. To avoid that it is suitable to design D mode to be proportional to the change of the output variable.

PD controller is often used in control of moving objects such are flying and underwater vehicles, ships, rockets etc. One of the reason is in stabilizing effect of PD controller on sudden changes in heading variable $y(t)$. Often a "rate gyro" for velocity measurement is used as sensor of heading change of moving object.



Block diagram of PD controller

P-I-D Controller:

P-I-D controller has the optimum control dynamics including zero steady state error, fast response (short rise time), no oscillations and higher stability. The necessity of using a derivative gain component in addition to the PI controller is to eliminate the overshoot and the oscillations occurring in the output response of the system. One of the main advantages of the P-I-D controller is that it can be used with higher order processes including more than single energy storage.

Proportional Integral Derivative (PID) Controller.

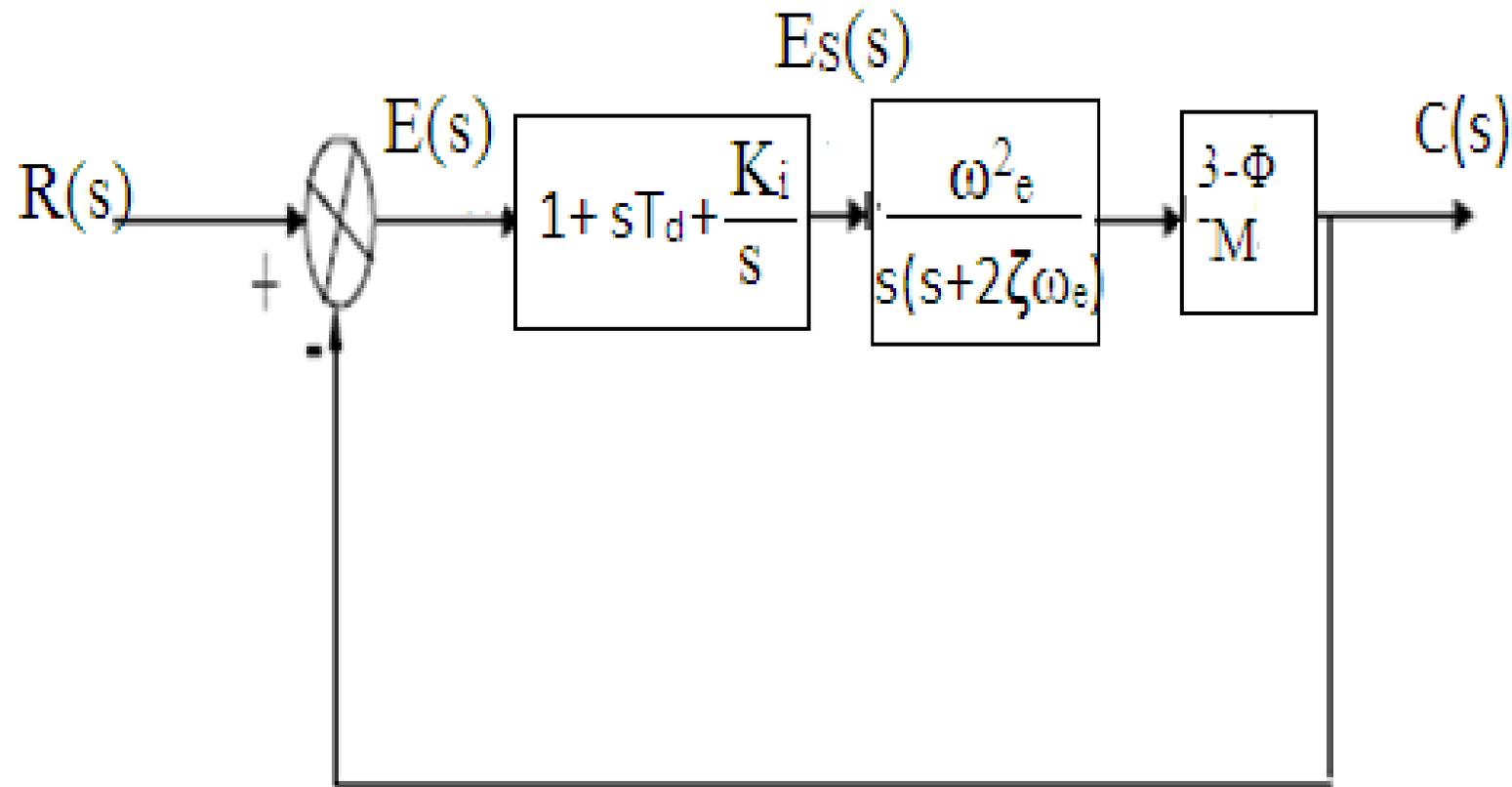
Many industrial controllers employ a proportional, integral plus differential PID regulator arrangement that can be tailored to optimize a particular control system. PID controller is most commonly used algorithm for controller design and it is most widely used controller in industry.

The controllers used in industry are either PID controller or its improved version. The basic types of PID controller are parallel controller, serial controller, and mixed controller.

The PID controller algorithm utilized for is design velocity algorithm, it is also called incremental algorithm. In the industry, PID controllers are the most common control methodology to use in real applications.

PID controller has all the necessary dynamics: fast reaction on change of the controller input (D mode), increase in control signal to lead error towards zero (I mode) and suitable action inside control error area to eliminate oscillations (P mode).

Derivative mode improves stability of the system and enables increase in gain K and decrease in integral time constant T_i , which increases speed of the controller response. PID controllers are the most often used controllers in the process industry. The majority of control systems in the world are operated PID controllers. It has been reported that 98% of the control loops in the pulp and paper industries are controlled by single-input single output PI controllers and that in process control applications, more than 95% of the controllers are of the PID type controller. PID controller combines the advantage of proportional, derivative and integral control action.



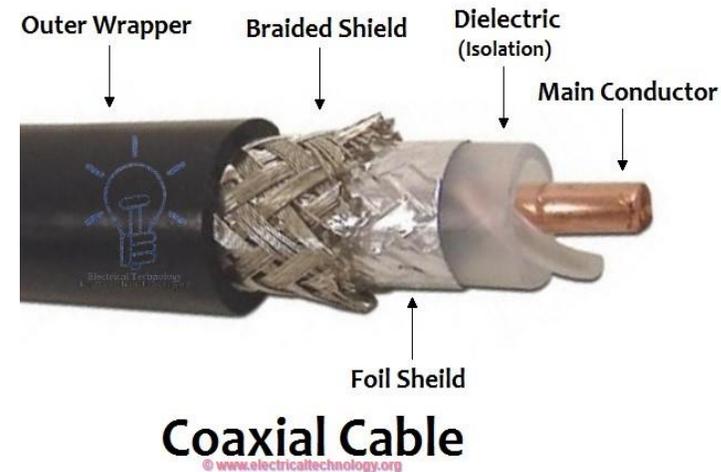
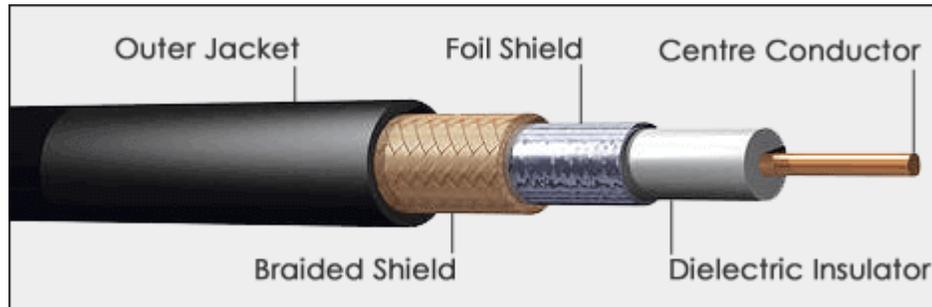
Block diagram of PID controller

$$u(t) = K_p \cdot e(t) + K_i \int e(t) \cdot dt + K_d \frac{de(t)}{dt}$$

The control signal is proportional to the error signal and the proportional gain K_p . A proportional controller will have the effect of reducing the rise time and will reduce, but never eliminate. If an integrator is added, the control signal is proportional to the integral of error and the integral gain K_i . Integral control will have the effect of reduced the error, in principle, to zero value. The in principle must be added, because there are always limits on accuracy in any system. Derivative control is used to anticipate the future behavior of the error signal by using corrective actions based on the rate of change in the error signal. The control signal is proportional to the derivative of the error and K_d is the derivative gain.

Communication Cables

- **Communication cables** are used for the transmission of data, voice and images, like *CCTV systems*.
- The first type of cables used for these functions were [coaxial cables](#).
- *Coaxial cables* have an inner conductor surrounded by a tubular insulating layer, surrounded by a tubular conducting shield, like is shown in **Figure**



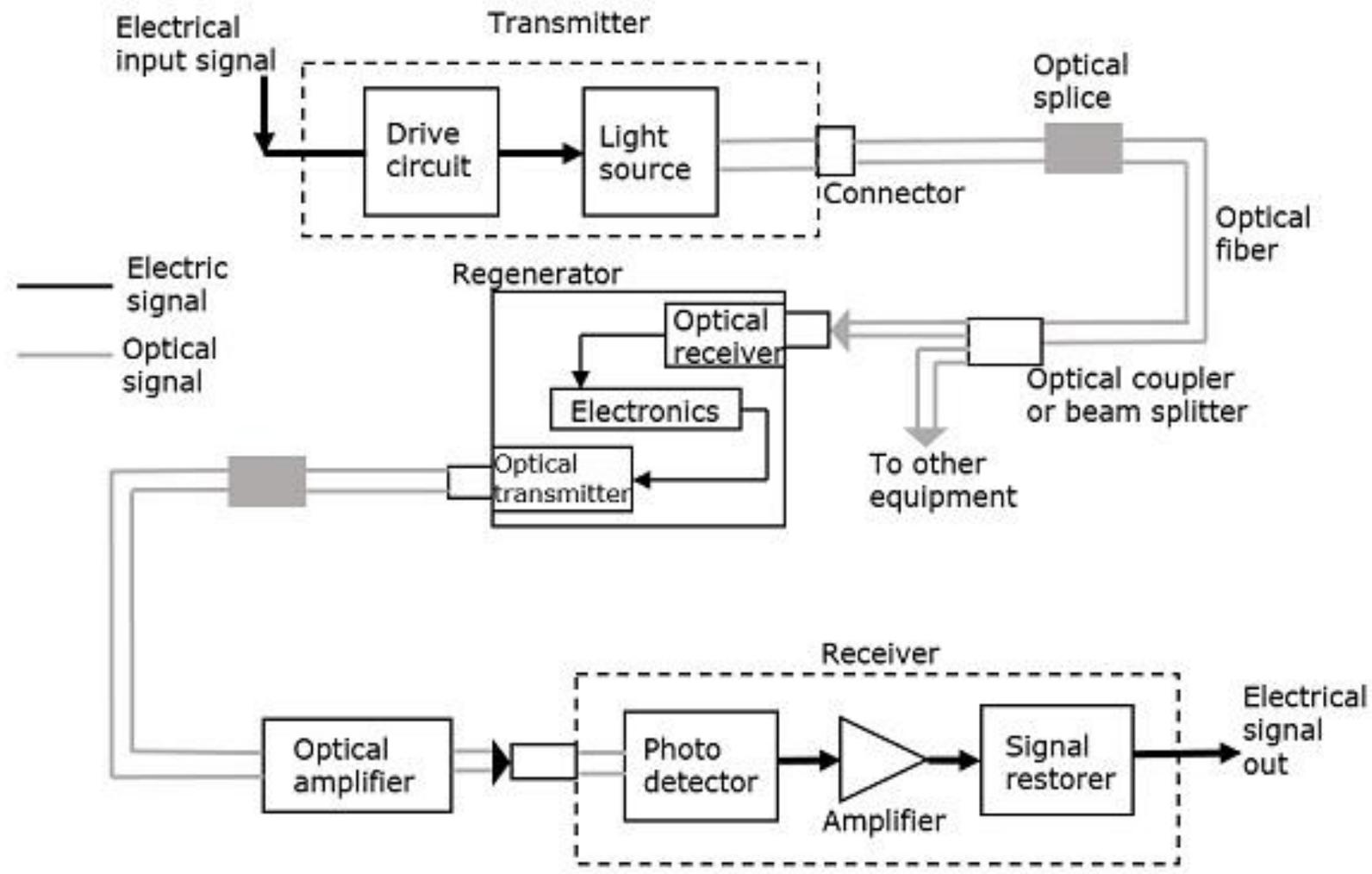
Types of Cables used in Control & Communication

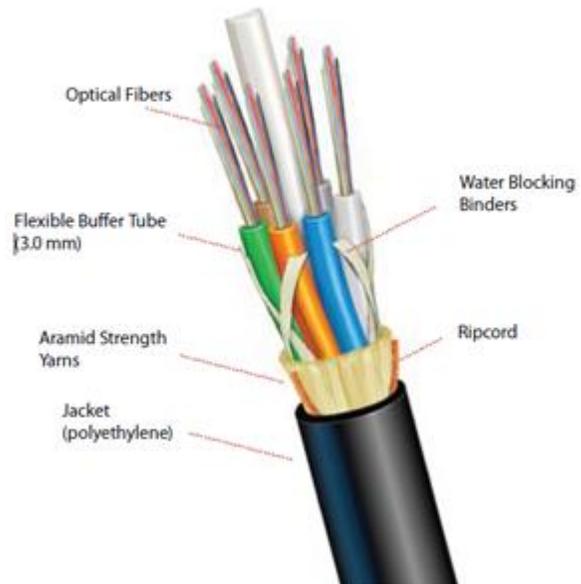
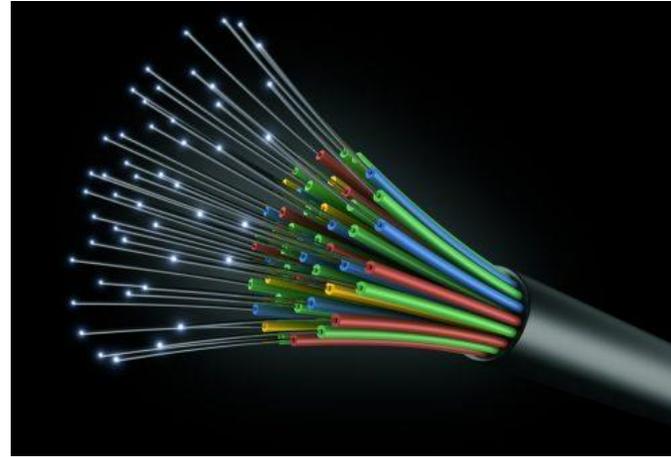
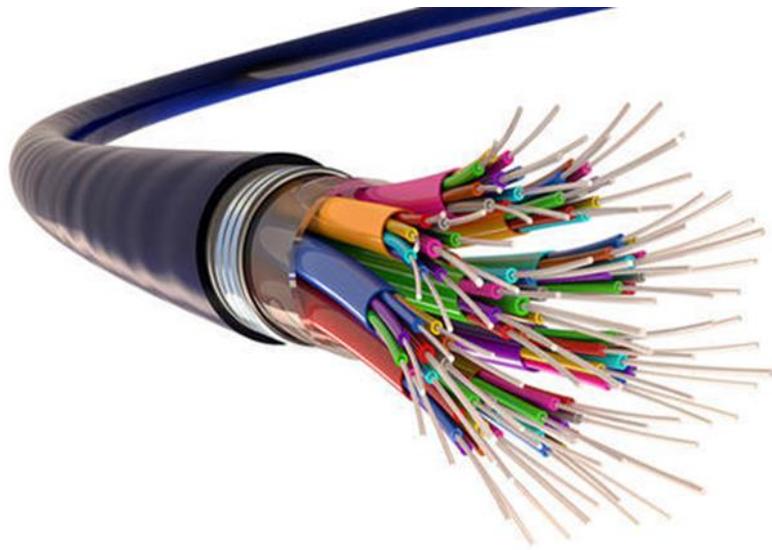
The three most common **types of communication cables** are

1. Twisted Pair,
2. Coaxial,
3. Fiber Optic.

Fiber Optic Cable

- Fiber optic cable consists of a bundle of glass threads, each of which is capable of transmitting message modulated onto light waves.
- Fiber Optic cable has a complex design and structure. This type of cable has an outer optical casing that surrounds the light and traps it within a central core. The inside of the cable (the core) can be configured in two different ways – Single-mode and multi-mode; although the difference may seem small, it makes a huge difference to the performance and usage of fiber optic cables.

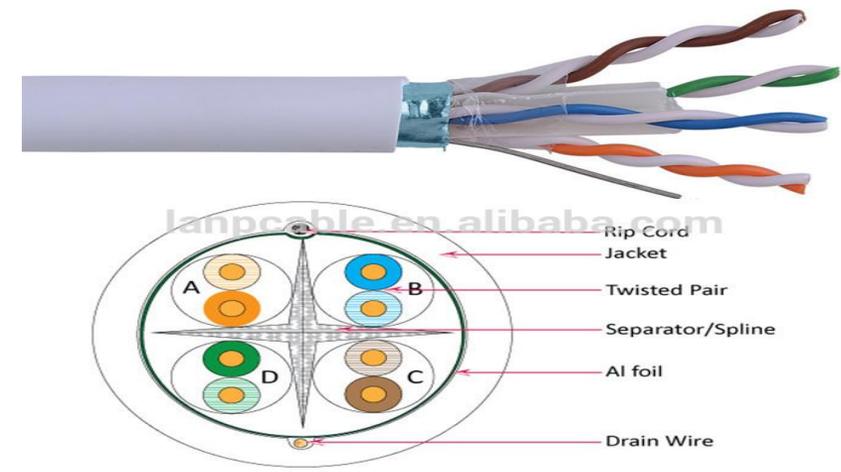
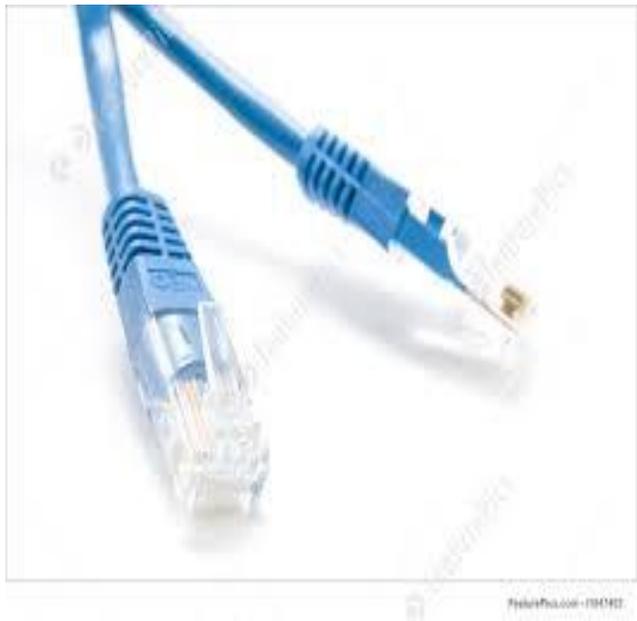
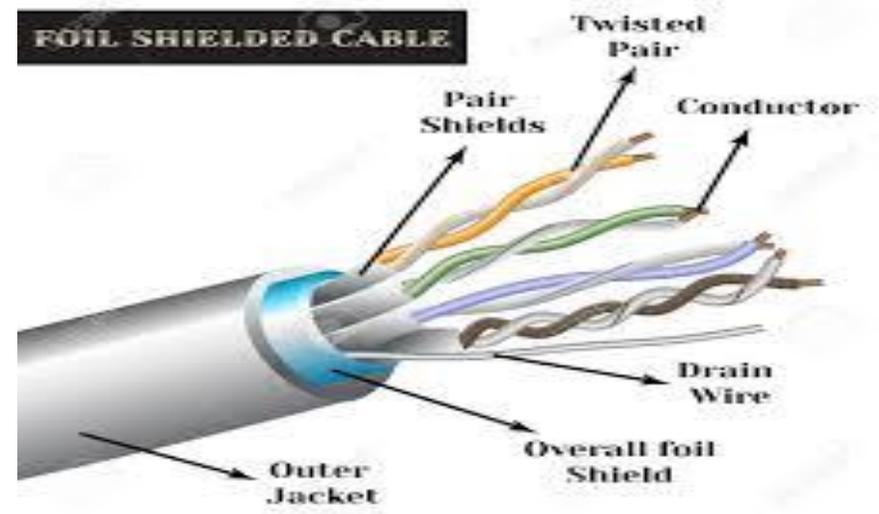
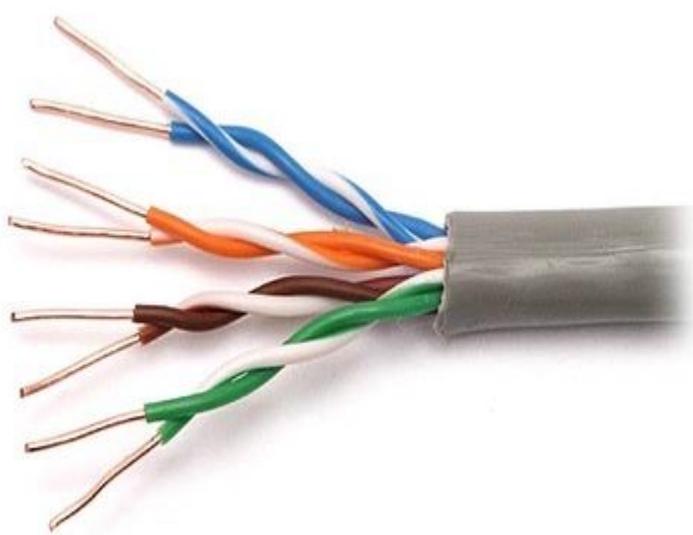




Twisted Pair Cable

Twisted pair cable is a type of ordinary wiring which connects home and many business computers to the telephone company. It is made by putting two separate insulated wires together in a twisted pattern and running them parallel to each other which helps to reduce crosstalk or electromagnetic induction between pairs of wires.

Twisted pair cable is good for transferring balanced differential signals. The practice of transmitting signals differentially dates back to the early days of telegraph and radio. The advantages of improved signal-to-noise ratio, crosstalk, and ground bounce that balanced signal transmission brings are particularly valuable in wide bandwidth and high fidelity systems.

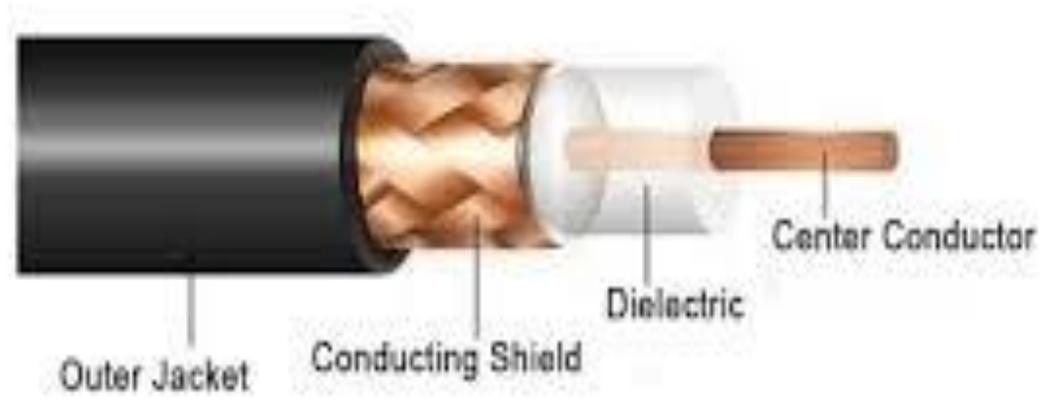


Coaxial Cable

Coaxial cable, or coax cable, is another type of copper cable which has an inner conductor surrounded by a foam insulation, symmetrically wrapped by a woven braided metal shield, then covered by in a plastic jacket.

This special design allows coaxial cable runs to be installed next to metal objects such as gutters without the power losses that occur in other types of transmission lines. Coaxial cable acts as a high-frequency transmission cables made up of a single solid-copper core and compared to twisted pair cable, it has 80 times or more the transmission capability.

This kind of cable is mainly adopted in feed lines connecting radio transmitters and receivers with their antennas, computer network connections, and distributing cable television signals.



Optical Fibers

Fiber optics (optical fibers) are long, thin strands of very pure glass about the size of a human hair. They are arranged in bundles called optical cables and used to transmit signals over long distances.

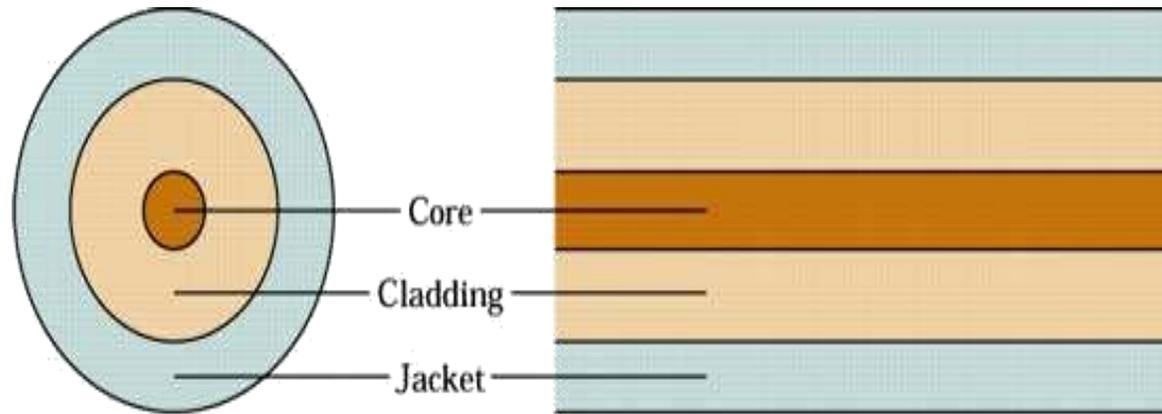
What are optical fibers

- Thin strands of pure glass
- Carry data over long distances
- At very high speeds
- Fiber can be bent or twisted

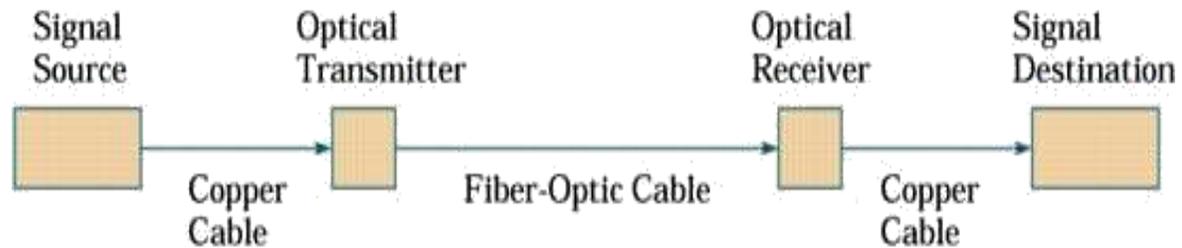
- An optical fiber is essentially a waveguide for light
- It consists of a **core** and **cladding** that surrounds the core
- The **index of refraction** of the cladding is less than that of the core, causing rays of light leaving the core to be refracted back into the core
- A light-emitting diode (LED) or **laser diode** (LD) can be used for the source

- Advantages of optical fiber include:
 - Greater bandwidth than copper
 - Lower loss
 - Immunity to **crosstalk**
 - No electrical hazard

Optical Fiber & Communications System



(a) Fiber cross section

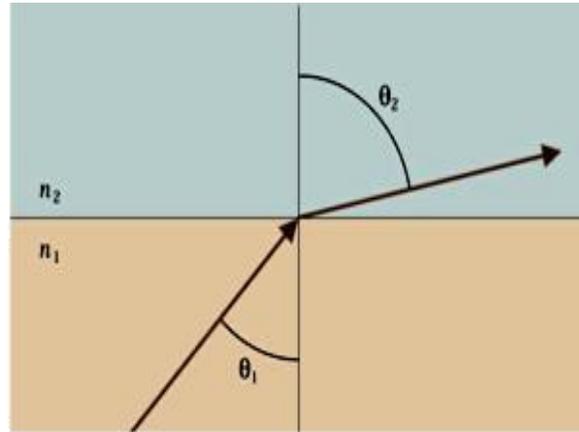


(b) System

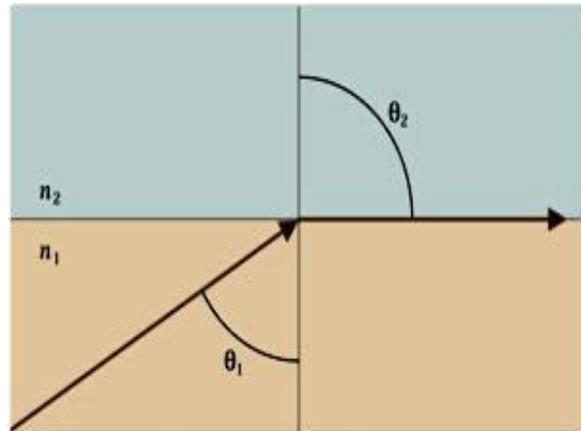
Total Internal Reflection

- Optical fibers work on the principle of **total internal reflection**
- With light, the refractive index is listed
- The **angle of refraction** at the interface between two media is governed by Snell's

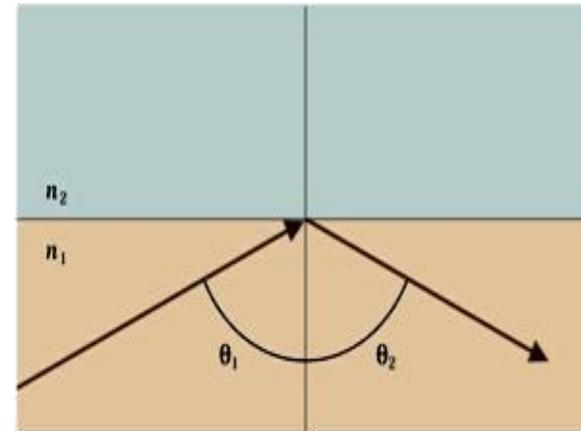
Refraction & Total Internal Reflection



(a) Angle of incidence less than critical angle



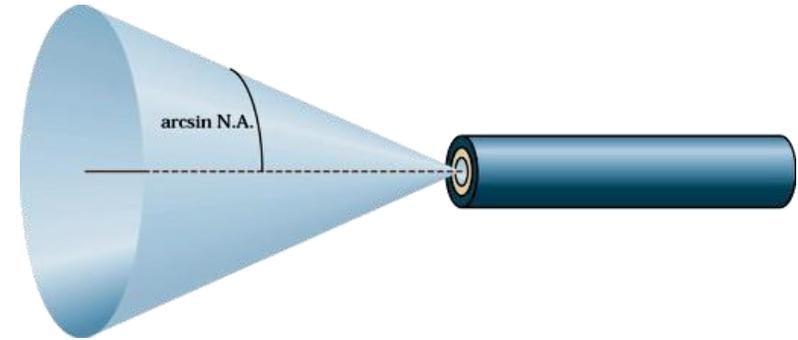
(b) Angle of incidence equal to critical angle



(c) Angle of incidence greater than critical angle

Numerical Aperture

- The **numerical aperture** of the fiber is closely related to the critical angle and is often used in the specification for optical fiber and the components that work with it
- The numerical aperture is given by the formula:
- The **angle of acceptance** is twice that given by the numerical aperture



Fibers can be bent!!

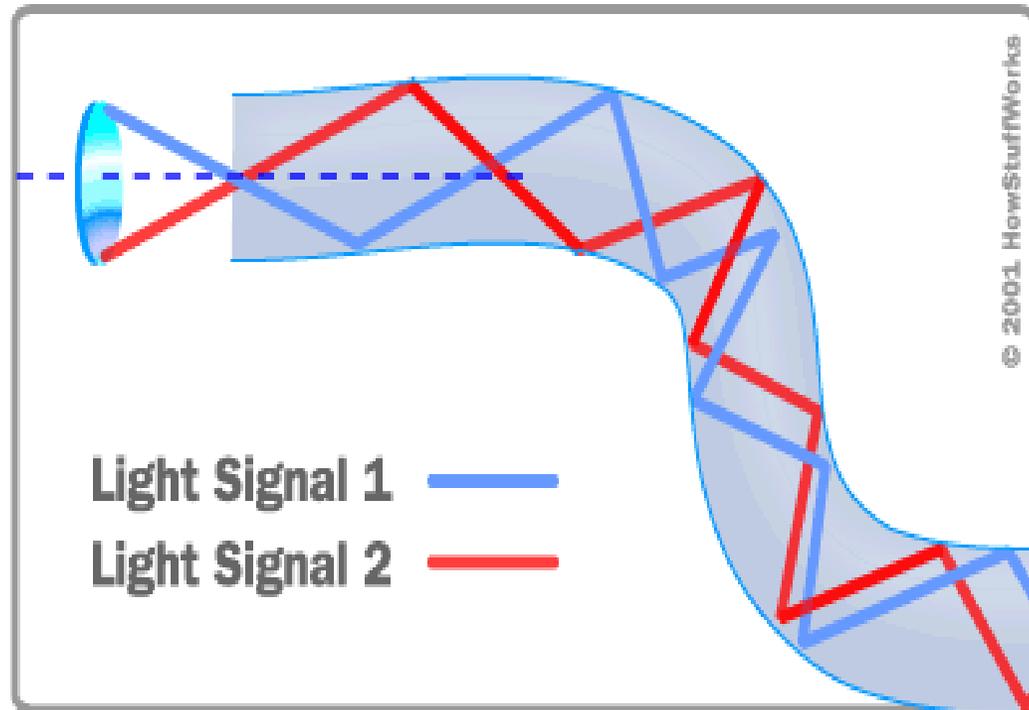


Fig: Illustration of total internal reflection

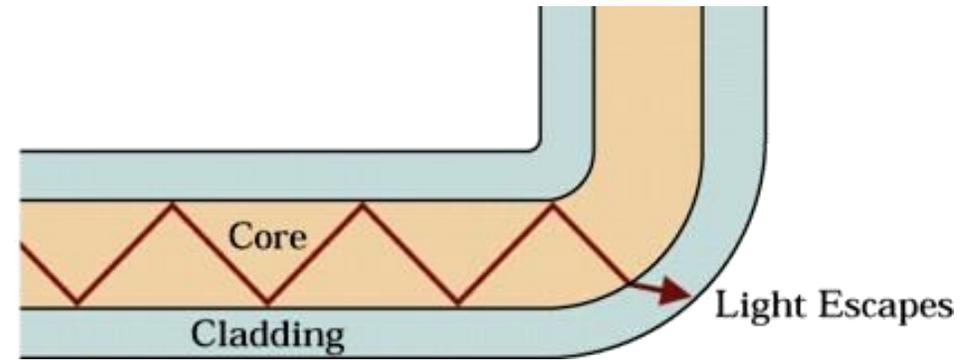
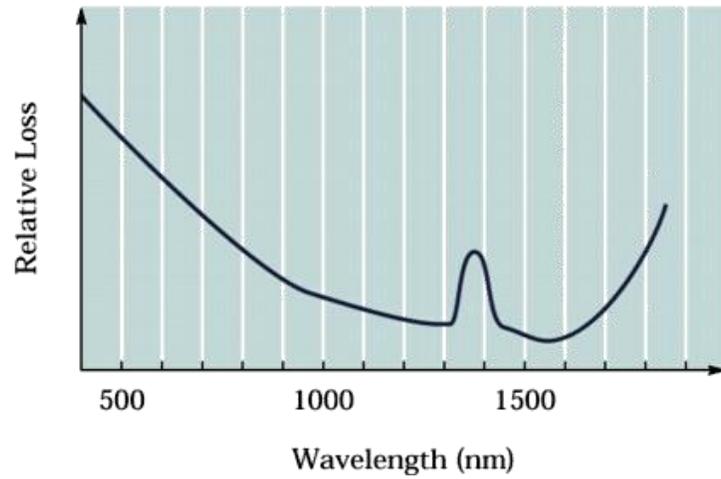
Types of optical fibers

- Single mode
 - only one signal can be transmitted
 - use of single frequency
- Multi mode
 - Several signals can be transmitted
 - Several frequencies used to modulate the signal

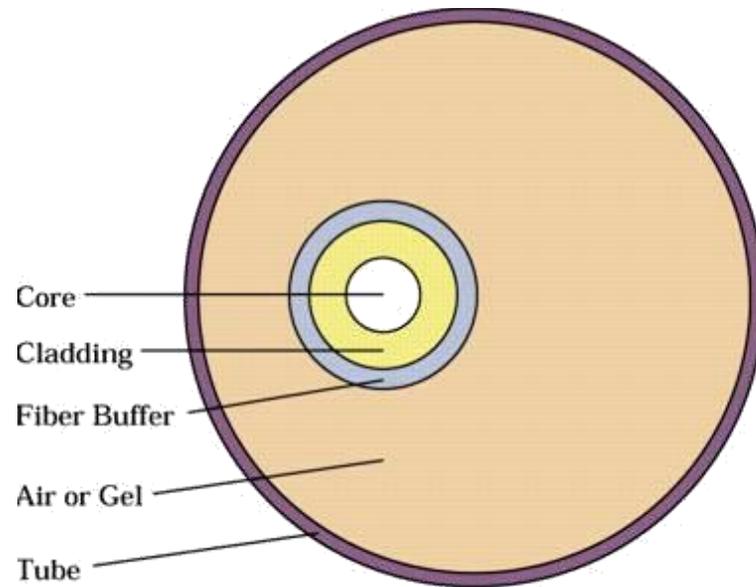
Losses in optical fibers

- Attenuation loss
- Dispersion loss
- Waveguide loss

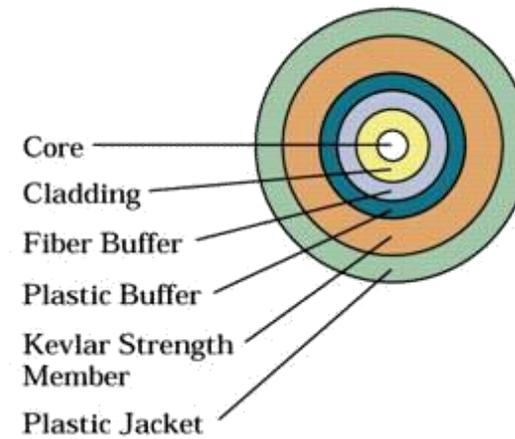
Types of Losses



Fiber-Optic Cable Construction



(a) Loose-tube construction



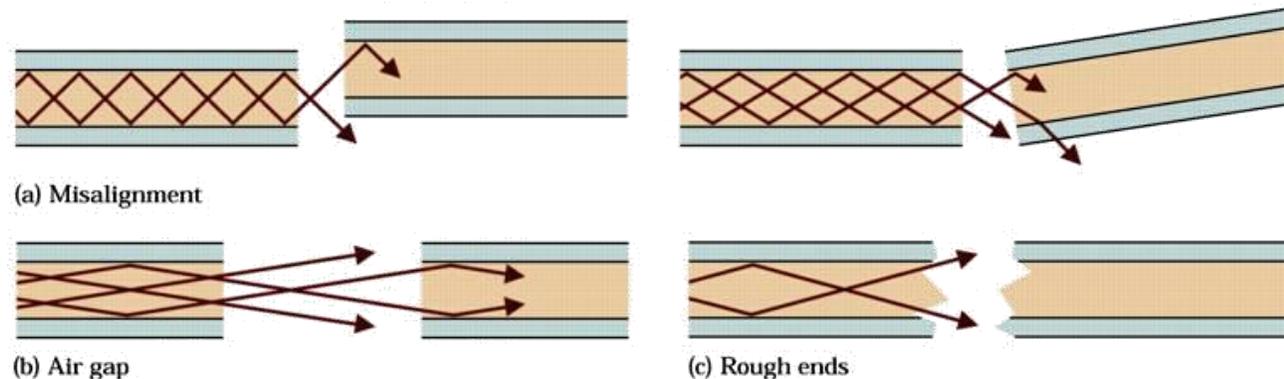
(b) Tight-buffer construction

Splices and Connectors

- To connect to fibers mechanically or by fusion
- Lot of signal loss possible
- Very accurate alignment necessary
- Most important cost factor
- Now being replaced by optical amplifiers

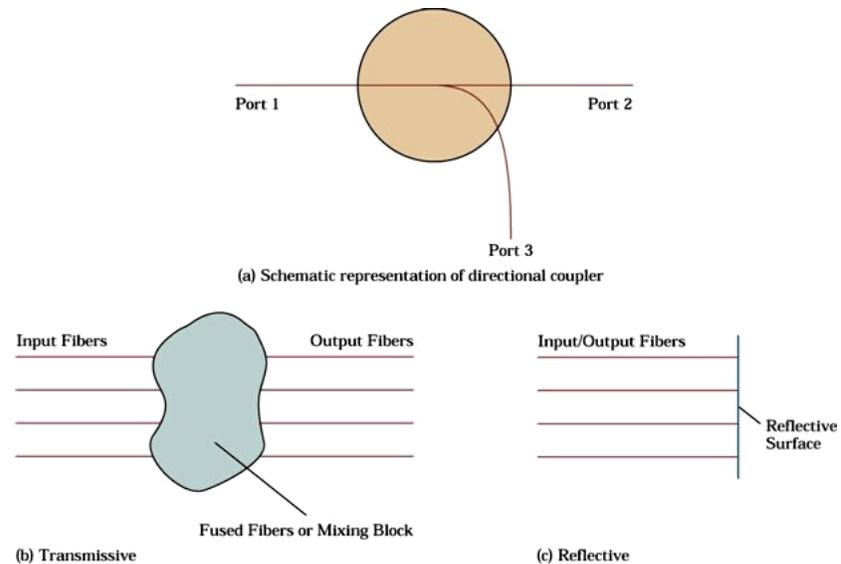
Splices and Connectors

- In fiber-optic systems, the losses from splices and connections can be more than in the cable itself
- Losses result from:
 - Axial or angular misalignment
 - Air gaps between the fibers
 - Rough surfaces at the ends of the fibers



Optical Couplers and Switches

- As with coaxial cable and microwave waveguides, it is possible to build power splitters and directional couplers for fiber-optic systems
- It is more complex and expensive to do this with fiber than with copper wire
- Optical couplers are categorized as either star couplers with multiple inputs and outputs or as tees, which have one input and two outputs

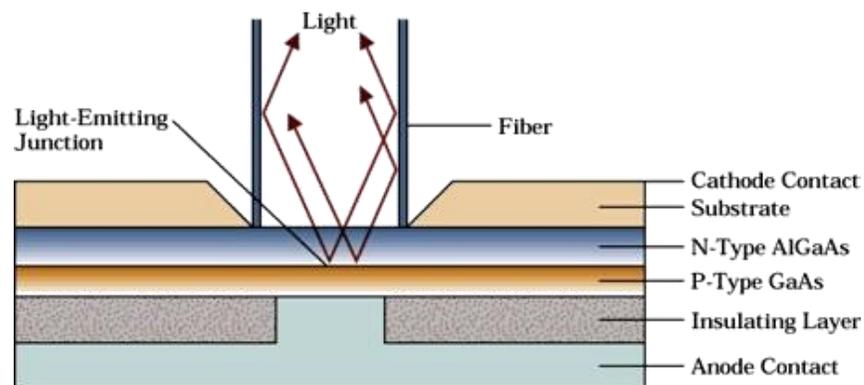


Optical Receivers

- Must be very sensitive
- Capable of picking up and amplifying signals of nanowatts
- Photodiodes and phototransistors
- These devices get 'turned ON' by light
- Produce photocurrent

Light-Emitting Diodes

- An LED is form of junction diode that is operated with forward bias
- Instead of generating heat at the PN junction, light is generated and passes through an opening or lens
- LEDs can be visible spectrum or infrared



Laser Diodes

- Laser diodes generate coherent, intense light of a very narrow bandwidth
- A laser diode has an emission linewidth of about 2 nm, compared to 50 nm for a common LED
- Laser diodes are constructed much like LEDs but operate at higher current levels

Fiber optic technology

- Sources
- Transmission medium
- Detectors

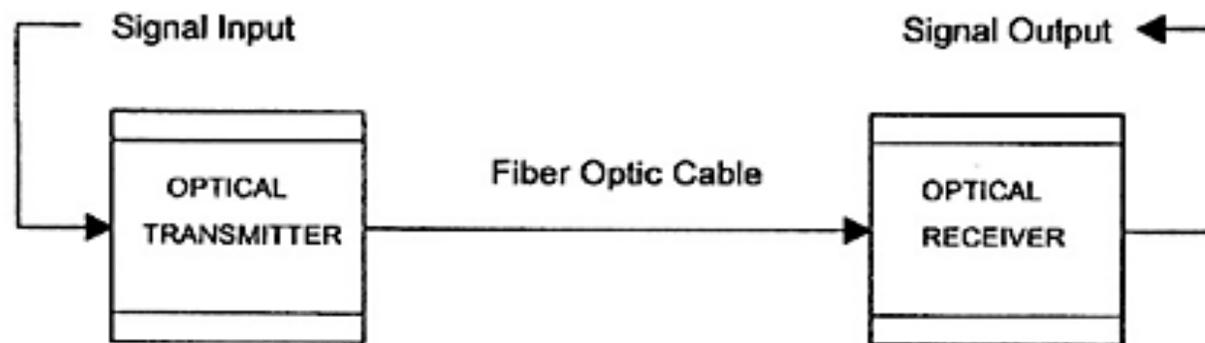
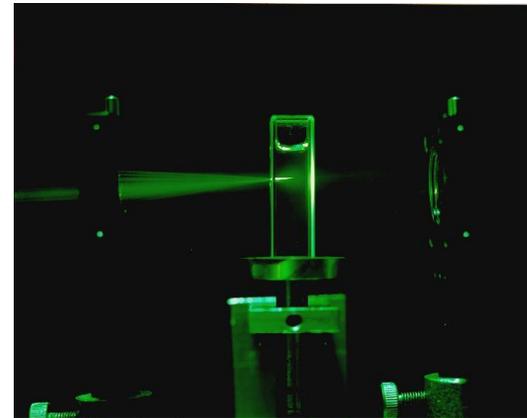
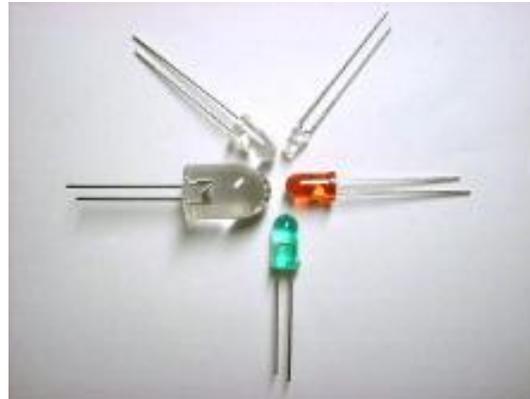


Fig: The fiber optic communication system

Sources of light

- Light emitting diodes
- Lasers

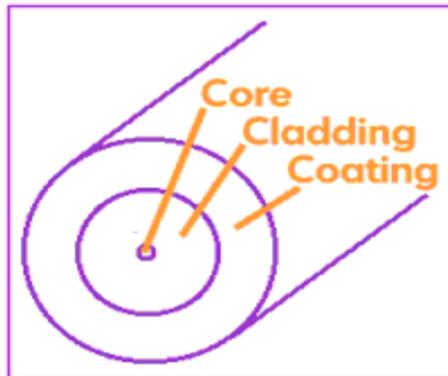


Sources

- Modulate electrical signals into optical signals
- Mostly modulate at 850nm, 1300nm and 1550 nm
- Lasers give high intensity, high frequency light
- LEDs are economical

Transmission medium

- Optical fiber is replacing copper
- Light is used as the carrier of information
- Much higher data rate



The optical fiber

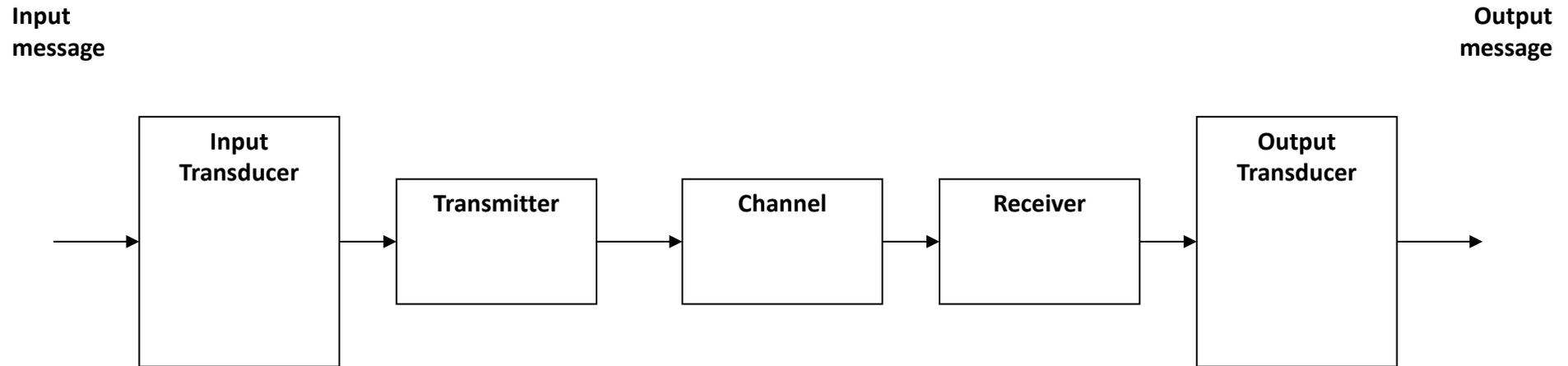
communication

The process of communication refers to the transmission or passage of information or message from the sender through a selected channel to the receiver overcoming barriers that affect its pace. The process of communication is a cyclic one as it begins with the sender and ends with the sender in the form of feedback.

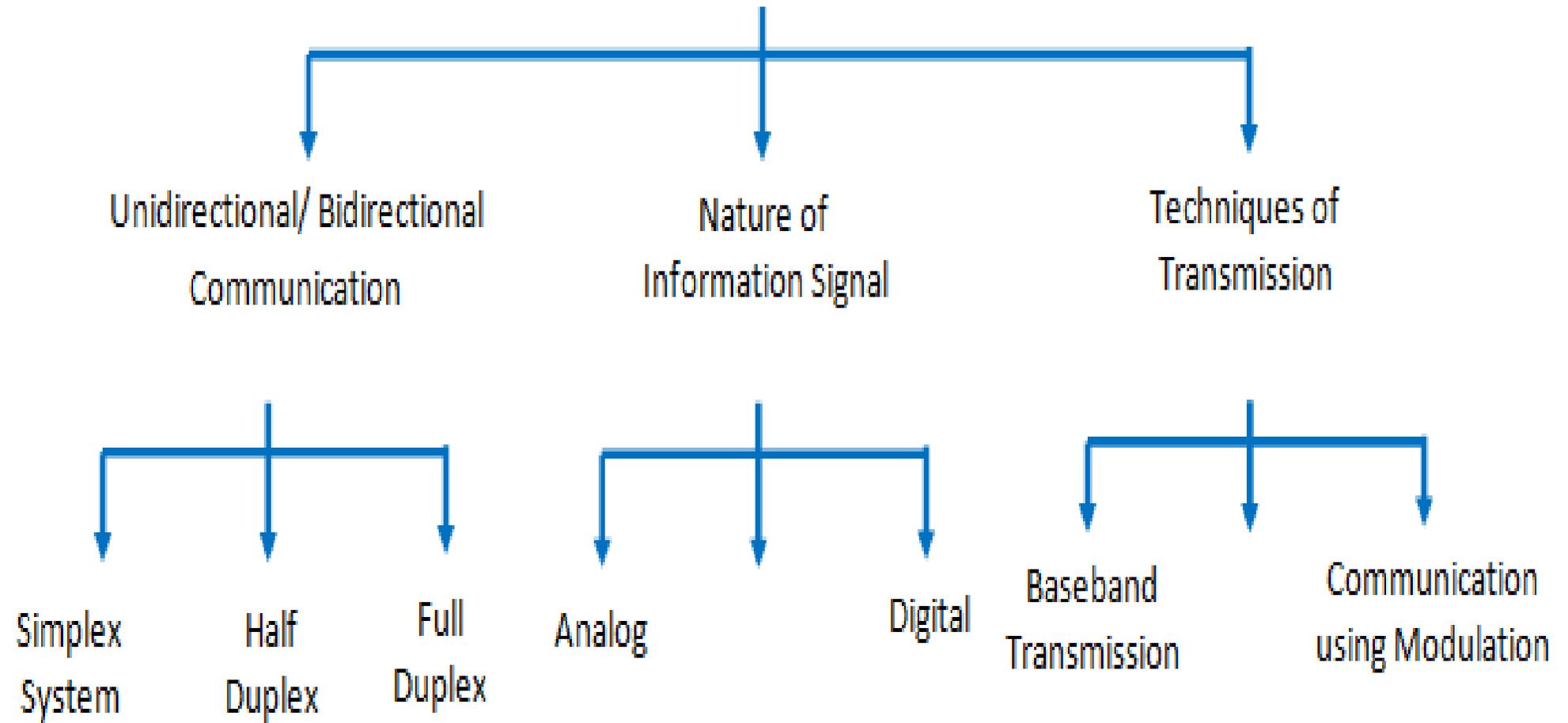
Seven major elements of communication process are:

- (1) sender
- (2) Ideas
- (3) encoding
- (4) communication channel
- (5) receiver
- (6) decoding and
- (7) feedback.

Elements of a Communication System



Electronic Communication System



Input Transducer: The message produced by a source must be converted by a transducer to a form suitable for the particular type of communication system.

*Example: In electrical communications, speech waves are **converted** by a microphone to voltage variation.*

Transmitter: The transmitter processes the input signal to produce a signal suits to the characteristics of the transmission channel.

*Signal **processing** for transmission almost always involves **modulation** and may also include **coding**. In addition to modulation, other functions performed by the transmitter are **amplification**, **filtering** and coupling the modulated signal to the channel.*

Channel: The channel can have different forms: The atmosphere (or free space), coaxial cable, fiber optic, waveguide, etc.

The signal undergoes some amount of degradation from noise, interference and distortion

Receiver: The receiver's function is to extract the desired signal from the received signal at the channel output and to convert it to a form suitable for the output transducer.

Other functions performed by the receiver: amplification (the received signal may be extremely weak), demodulation and filtering.

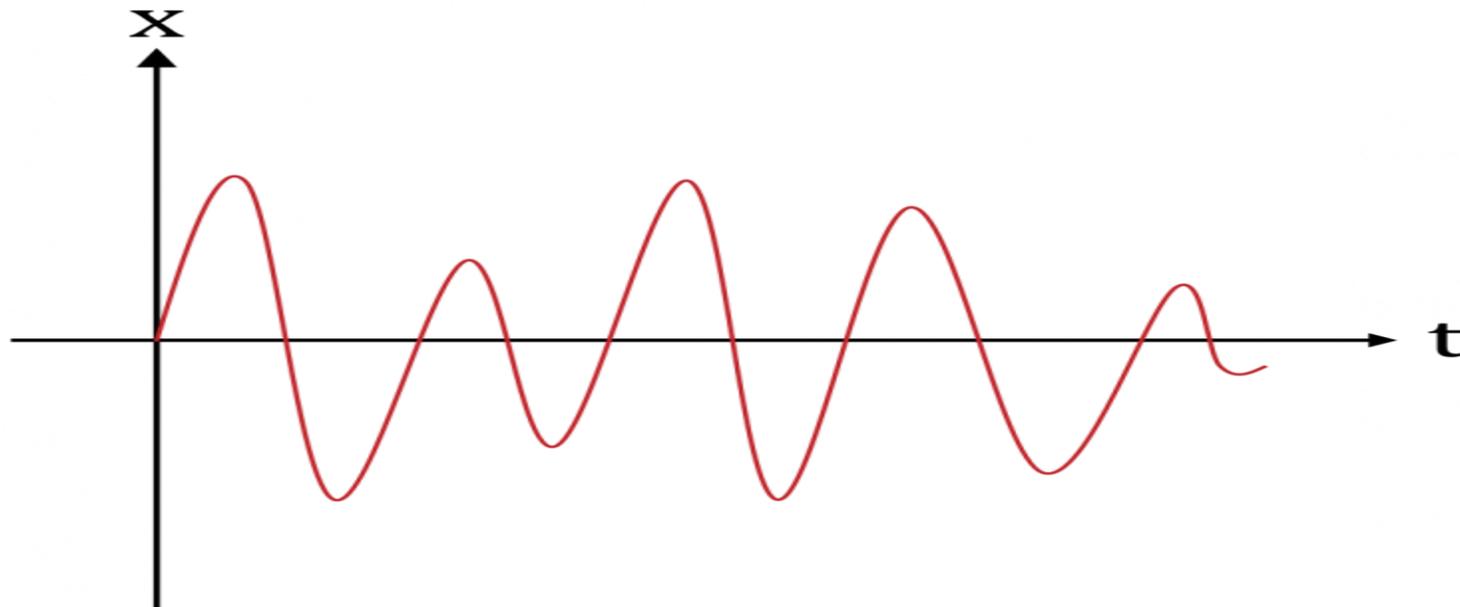
Output Transducer: Converts the electric signal at its input into the form desired by the system user.

Example: Loudspeaker, personal computer (PC), tape recorders.

What is Analog and Digital Signal?

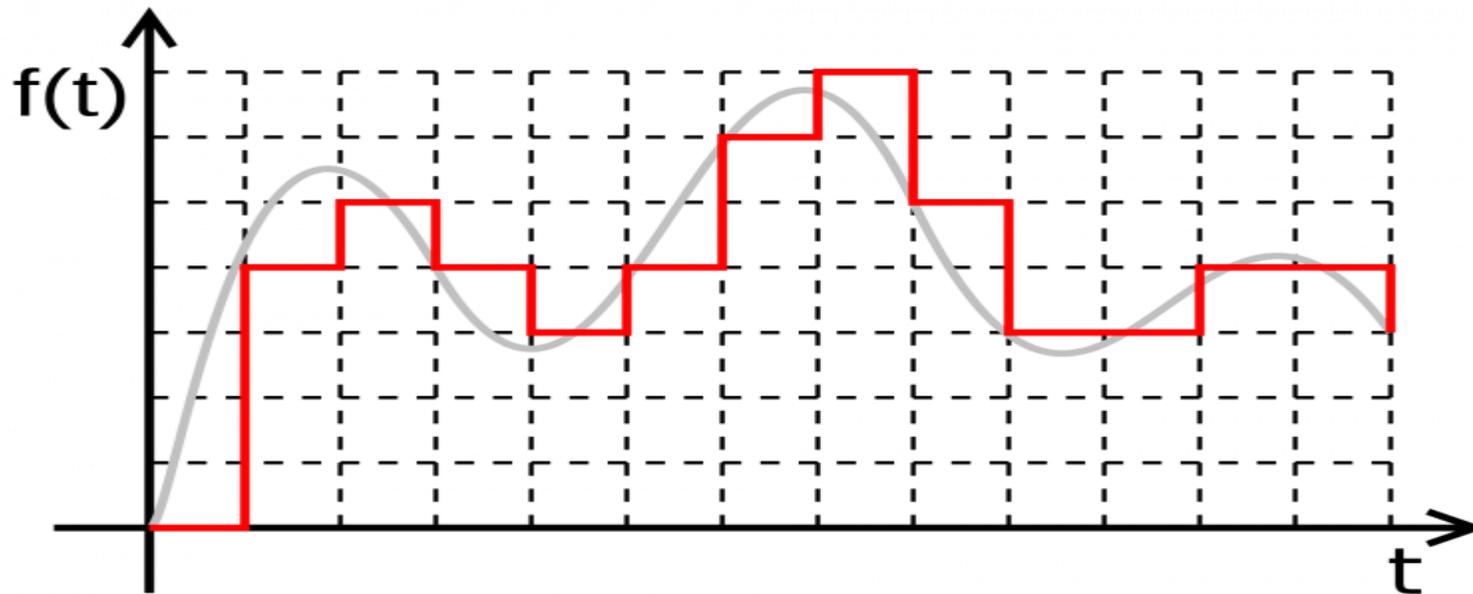
Analog Signal:

An analog signal is one type of continuous time-varying signals, and these are classified into composite and simple signals. A simple type of analog signal is nothing but a sine wave, and that can't be decomposed, whereas a composite type analog signal can be decomposed into numerous sine waves. An analog signal can be defined by using amplitude, time period otherwise frequency, & phase. Amplitude streaks the highest height of the signal, frequency streaks the rate at which an analog signal is varying, and phase streaks the signal position with respect to time nothing. An analog signal is not resistant toward the noise, therefore; it faces distortion as well as reduces the transmission quality. The analog signal value range cannot be fixed.



Digital Signal:

digital signals carry the data although it is a bit different. These signals are discrete or not continuous. A digital signal carries the data in the form of binary because it signifies in the bits. These signals can be decomposed into sine waves which are termed as harmonics. Every digital signal has amplitude, frequency, & phase like the analog signal. This signal can be defined by bit interval as well as bit rate. Here, bit interval is nothing but the required time for transmitting an only bit, whereas the bit rate is bit interval frequency.



Characteristics of Analog and Digital Signals

- The analog and Digital signal characteristics mainly include adaptability, continuity, representation, data type, signal type, medium of transmission, type of values, security, bandwidth, hardware, data storage, portability, data transmission, impedance, power consumption, recording data, use, rate of data transmission, examples and applications.

Adaptability

- Analog signals are less adjustable for a range of use, whereas digital signals are more adjustable for a range of use.

Continuity

- Analog signals use a continuous variety of amplitude values whereas digital signal takes a limited set of distinct values at consistently spaced spots in the time.

Type of Data

- Analog signals are continuous in nature, whereas digital signals are discrete.

Type of Waves

Analog signal wave type is sinusoidal, whereas a digital signal is a square wave.

Medium of Transmission

Analog signal medium of transmission is wire or wireless, whereas a digital signal is a wire.

Type of Values

Analog signal value type is positive as well as negative, whereas a digital signal is positive.

Security

The security of an analog signal is not encrypted, whereas a digital signal is encrypted.

Bandwidth

The analog signal bandwidth is low, whereas the digital signal is high.

Hardware

Analog signal hardware is not elastic, whereas digital is elastic in execution

Data Storage

- The data storage of an analog signal is in the wave signal form, whereas digital signal stores the data in the binary bit form.

Portability

- Analog signals are portable similar to the thermometer and low cost, whereas digital signals are portable similar to computers and expensive.

Data transmission

- In analog, the signal can be deterioration due to noise throughout transmission, whereas digital signal can be noise resistant throughout transmission devoid of any deterioration.

Impedance

- The impedance of the analog signal is low, whereas the digital signal is high.

Power Consumption

- Analog devices use more power, whereas digital devices use less power.

Data Transmission Rate

- The data transmission rate in the analog signal is slow, whereas in the digital signal it is faster.

Examples

- The best examples of the analog signal are video, human voice in the air, radio transmission waves or TV transmission waves.

Applications

- Analog signals can be utilized in analog devices exclusively, thermometer, whereas digital signals are appropriate for digital electronic devices like computers, PDA, cell phones.

- **bandwidth** refers to how much digital information we can send or receive across a connection in a certain amount of time. Sometimes it's called **data transfer rate** too.
- Most of the time, bandwidth refers to maximum throughput, and the information that is sent through is expressed in bits per second (a bit is the smallest unit of digital data that is represented as a 1 or 0). Since the number of bits can be a really large number, we might use a term like kilobits per second (Kbps or 1,000 bits per second) or megabits per second (Mbps or 1,000,000 bits per second) for how many bits can be sent or received in a second. An example of a common internet speed is around 10Mbps (megabits per second), which works out to about 1.3MBps (megabytes per second). That's pretty fast!
- By the way, if you're wondering how 10Mbps works out to approximately 1.3MBps, simply divide by 8, which is the number of bits in each byte.
- Bear in mind that this kind of bandwidth is used to describe networking. Bandwidth can also refer to processor bandwidth, which is measured in hertz. So, if you might sometimes feel confused, just remember that network bandwidth usually refers to maximum bits per second.