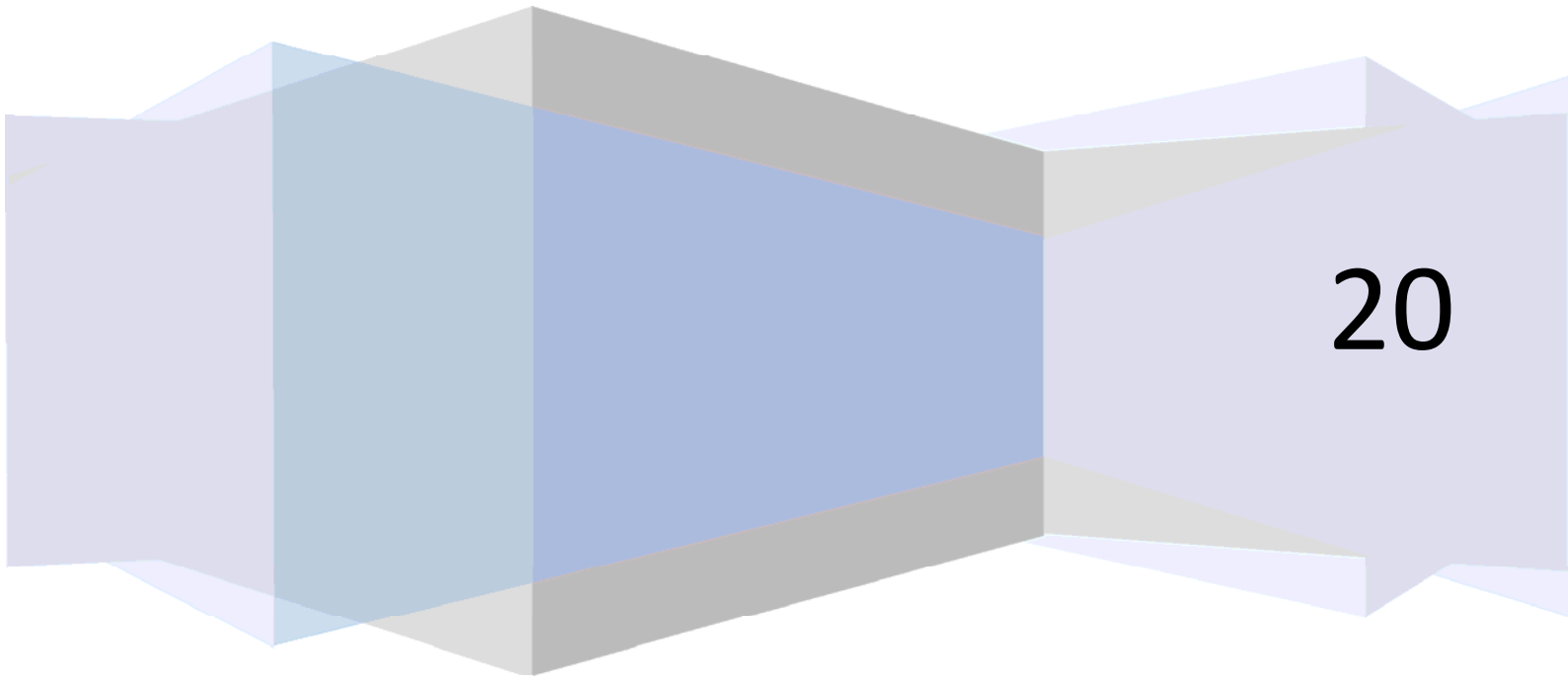
SMIT DIPLOMA ENGG. , ANKUSHPUR

STUDY MATERIAL ON

EMBEDDED SYSTEM, MICROCONTROLLER AND PLC

6TH SEM, ETC&TC



20

PROGRAMMABLE LOGIC CONTROLLERS

INTRODUCTION:-

A programmable logic controller (PLC) is a specialized computer used to control machines and processes. It uses a programmable memory to store instructions and execute specific functions that include on/off control, Timing, Counting, Sequencing, arithmetic and data handling. Programmable logic controllers are used for the control and operation of manufacturing process equipment and machinery.

PARTS OF A PLC:-

- A typical PLC can be divided into parts as illustrated in the Figure below. These components are the central processing unit (CPU), the input /output (I/O) section, the power supply and the programming device.
- The term architecture can refer to PLC hardware to PLC software, or to a combination of both.

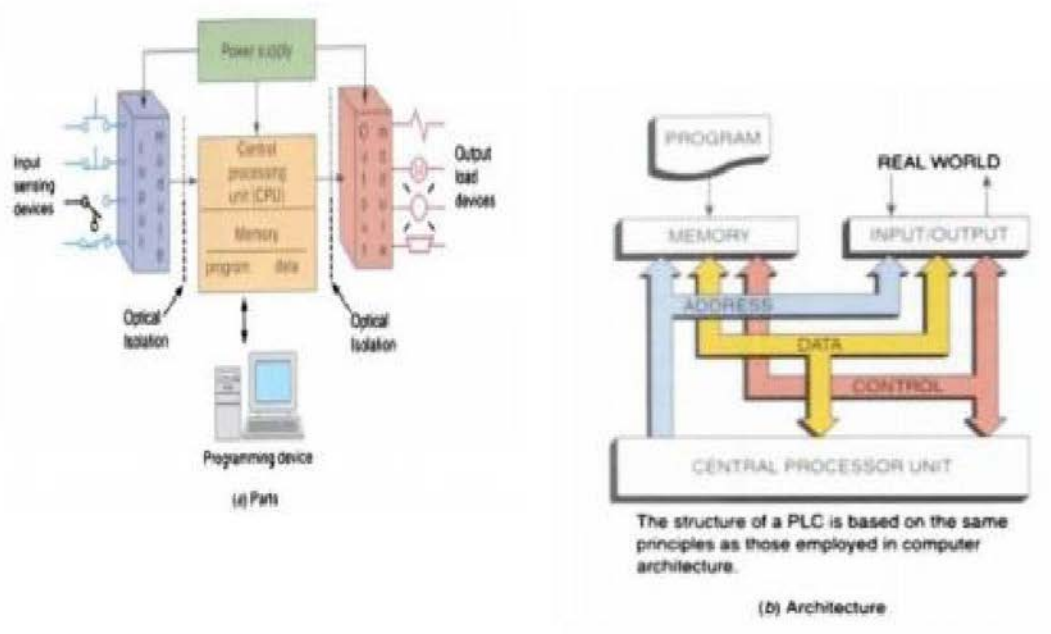


FIGURE: - PLC parts and architecture

- There are two ways in which I/O is incorporated into the PLC: fixed and modular.
- Fixed I/ O is typical of small PLCs that come in one package with no separate, removable units. The processor and I/O are packaged together, and the I/O terminals are available but cannot be changed.
- The main advantage of this type of packaging is lower cost.
- One disadvantage of fixed I/O is its lack of flexibility.

- Modular I/O in figure shown above is divided by compartments into which separate modules can be plugged. This feature greatly increases your options and the unit's flexibility. The basic modular controller consists of a rack, power supply, processor module (CPU), input/output (l /O modules), and an operator interface for programming and monitoring. The modules plug into a rack. When a module is slid into the rack, it makes an electrical connection with a series of contacts called the backplane, located at the rear of the rack.
- The power supply supplies dc power to other modules that plug into the rack. For large PLC systems, this power supply does not normally supply power to the field devices. With larger systems, power to field devices is provided by external alternating current (ac) or direct current (dc) supplies.
- The processor (CPU) is the "brain" of the PLC. A processor usually consists of a microprocessor for implementing the logic and controlling the communications among the modules. The processor requires memory for storing the results of the logical operations performed by the microprocessor. Memory is also required for the program EPROM or EEPROM plus RAM.
- The I/O section consists of input modules and output modules (Fig. shown below). The I/O system forms the interface by which field devices are connected to the controller. The purpose of this interface is to condition the various signals received from or sent to external field devices. Input devices such as pushbuttons, limit switches, sensors, selector switches, and thumbwheel switches are hardwired to terminals on the input modules. Output devices such as small motors, motor starters, solenoid valves, and indicator lights are hardwired to the terminals on the output modules.

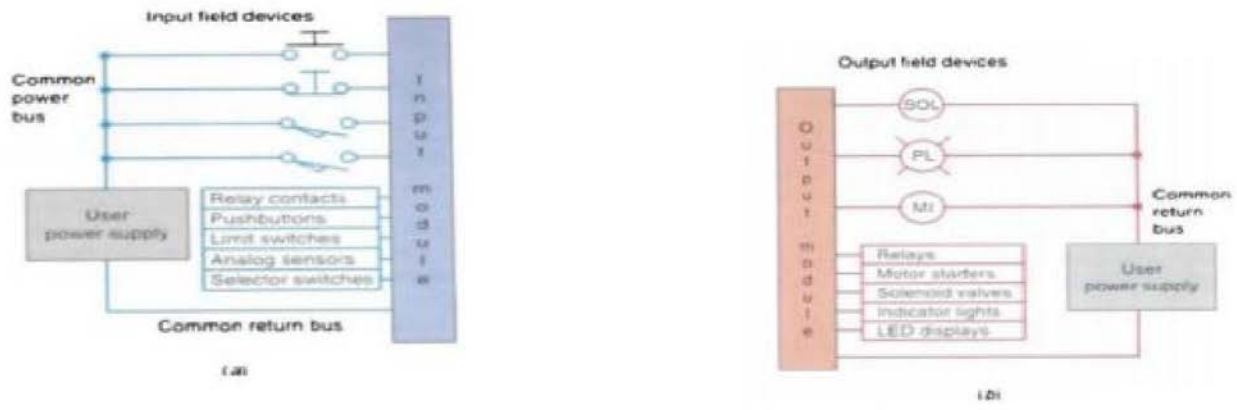


FIGURE: (a) Typical input module (b) typical output module

- The programming device, or terminal, is used to enter the desired program into the memory of the processor. This program is entered using relay ladder logic, which is the most popular programming language used by all major manufacturers of PLCs. Ladder logic programming language uses instead of words, graphic symbols that show their intended outcome. It is a special language written to make it easy for people familiar with relay logic control to program the PLC.

THE I/O SECTION:-

- The I/O section consists of an I/O rack and individual I/O modules similar to that shown in Figure shown below. Input interface modules accept signals from the machine or process devices and convert them into signals that can be used by the controller. Output interface modules convert controller signals into external signals used to control the machine or process.

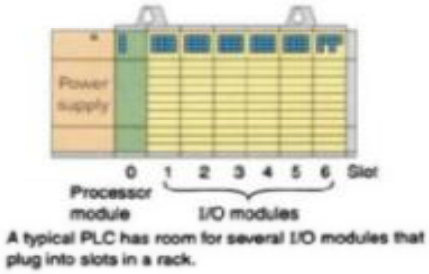
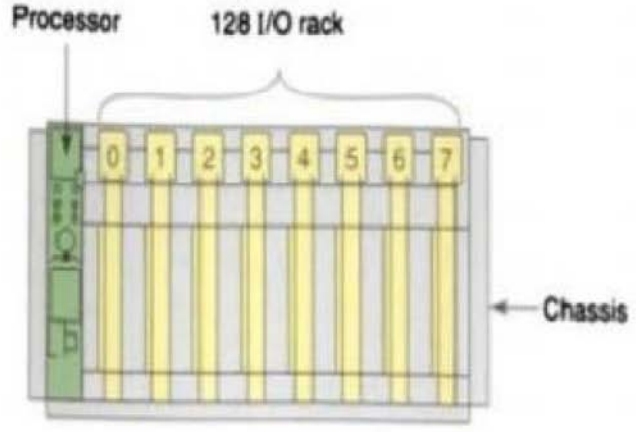
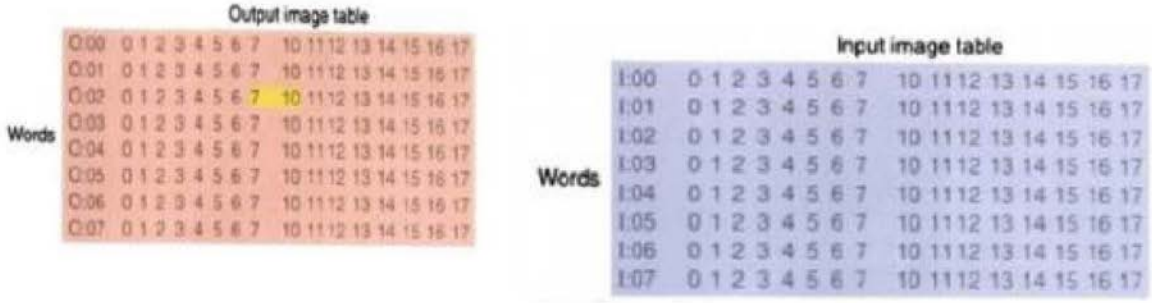


Figure: I/O section

- The I/O system provides an interface between the hardwired components in the field and the CPU. The input interface allows status information regarding processes to be communicated to the CPU, and thus allows the CPU to communicate operating signals through the output interface to the process devices under its control.

- A chassis is a physical hardware assembly that houses devices such as I/O modules, processor modules, and power supplies. Chassis come in different sizes according to the number of slots they contain. In general, they can have 4, 8, 12, or 16 slots.
- A logical rack is an addressable unit consisting of 128 input points and 128 output points. A rack uses 8 words in the input image table file and 8 words in the output image table file. A word in the output image table file and its corresponding word in the input image table file are called an I/O group. A rack can contain a maximum of 8 I/O groups (numbered from 0 through 7) for up to 128 discrete I/O (Fig. shown below). There can be more than one rack in a chassis and more than one chassis in a rack.



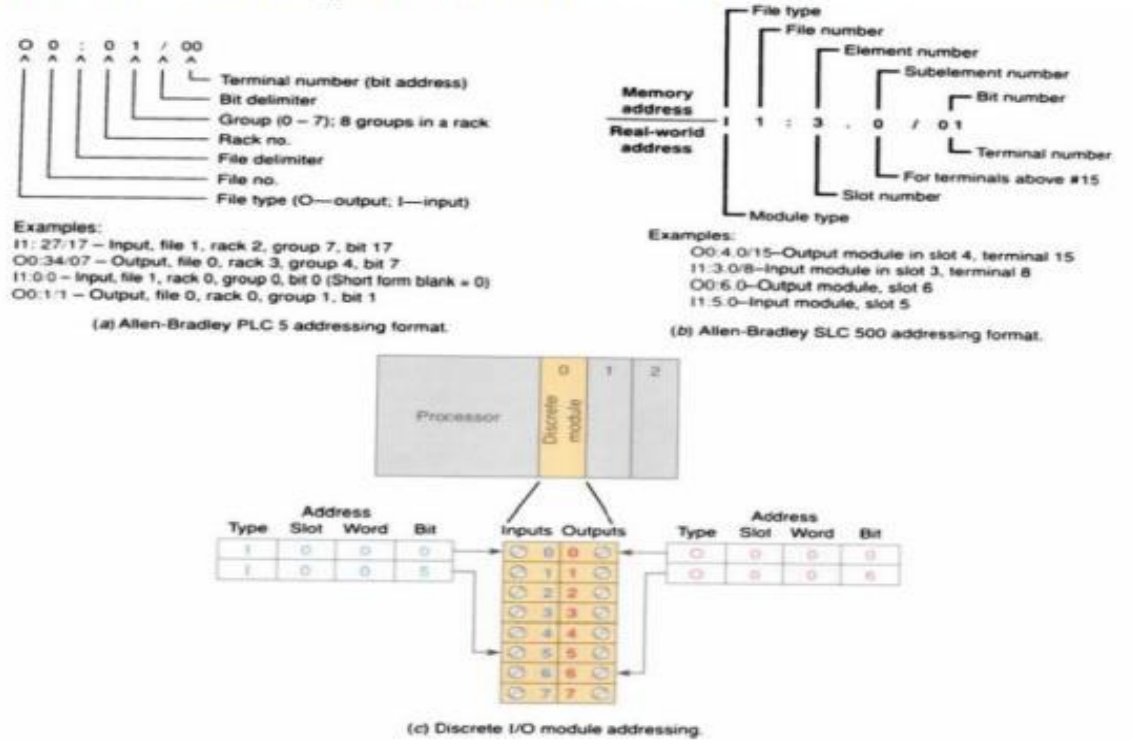


Allen-Bradley PLC-5 family uses an octal numbering system to address bit locations. Notice that the 16 bits are numbered 0 through 7 and 10 through 17. In the octal numbering system, the numbers 8 and 9 are never used.

Figure: Logical rack

- One benefit of a PLC system is the ability to locate the I/O modules near the field devices to minimize the amount of wiring required. This rack is referred to as a remote rack when it is located away from the processor module.
- The location of a module within a rack and the terminal number of a module to which an

input or output device is connected will determine the device's address (Fig. shown below). Each input and output device must have a specific address. This address is used by the processor to identify where the device is located to monitor or control it. In addition there is some means of connecting field wiring on the I/O module housing. Connecting the field wiring to the I/O housing allows easier disconnection and reconnection of the wiring to change modules. Lights are also added to each module to indicate the ON or OFF status of each I/O circuit. Most output modules also have blown fuse indicators.



In general, basic addressing elements include:

- **TYPE**

  The type determines if an input or output is being addressed.
- **SLOT**

  The slot number is the physical location of the I/O module. This may be a combination of the rack number and the slot number when using expansion racks.
- **WORD AND BIT**

  The word and bit are used to identify the actual terminal connection in a particular I/O module. A discrete module usually uses only one word, and each connection corresponds to a different bit that makes up' the word.

PLC MEMORY ORGANIZATION:-

- The term processor memory organization refers to how certain areas of memory in a given PLC are used.

Figure given below shows an illustration of the Allen-Bradley PLC-2 memory organization, known as a memory map. Every PLC has a memory map, but it may not be like the one illustrated. The memory space can be divided into two broad categories: the user program and the data table. The individual sections, their order, and the sections' length will vary and may be fixed or variable, depending on the manufacturer and model.

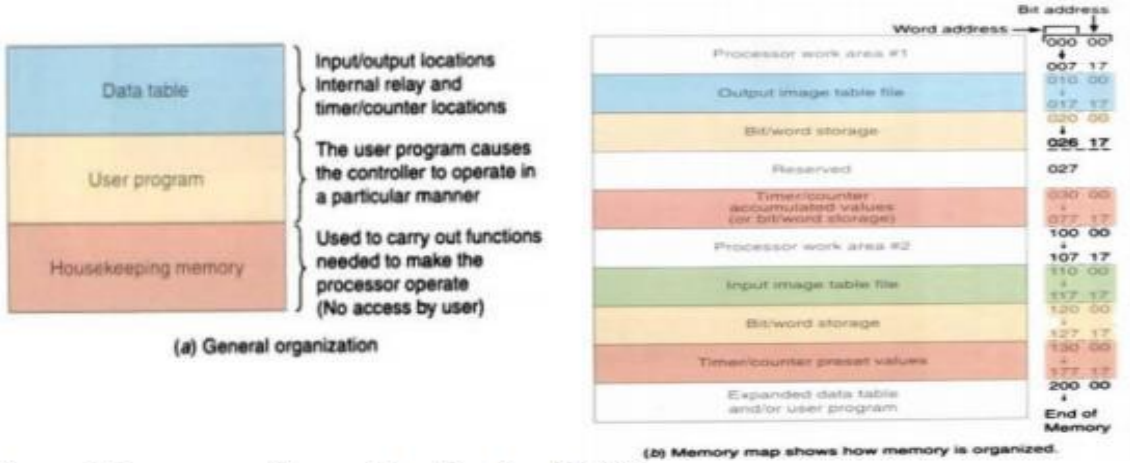


Figure: Memory map for an Allen-Bradley PLC-2

- The user program is where the logic ladder program is entered and stored. The user program will account for most of the total memory of a given PLC system. It contains the logic that controls the machine operation. This logic consists of instructions that are programmed in a ladder logic format. Most instructions require one word of memory.
- The data table stores the information needed to carry out the user program. This includes information such as the status of input and output devices, timer and counter values, data storage, and so on. Contents of the data table can be divided into two categories: status data and numbers or codes. Status is ON/OFF type of information represented by 1s and 0s, stored in unique bit locations. Number or code information is represented by groups of bits that are stored in unique byte or word locations.
- A processor file is the collection of program files and data files created under a particular processor file name. It contains all the instructions, data, and configuration information pertaining to a user program.

Figure shown below shows typical program and data file memory organization for an Allen-Bradley SLC-500 controller. The contents of each file are outlined in the sections that follow.

Program files:-

Program files are the areas of processor memory where ladder logic programming is stored. They may include:

- **System functions (file 0) -** This file is always included and contains various system-related information and user programmed information such as processor type. I/O configuration processor files name, and password.
- **Reserved (file 1)**-This file is reserved by the processor and is not accessible to the user.
- **Main ladder program (file 2)** - This file is always included and contains user programmed instructions that define how the controller is to operate.
- **Subroutine ladder program (files3- 255)**-These files are user-created and are activated according to subroutine instructions residing in the main ladder program file.

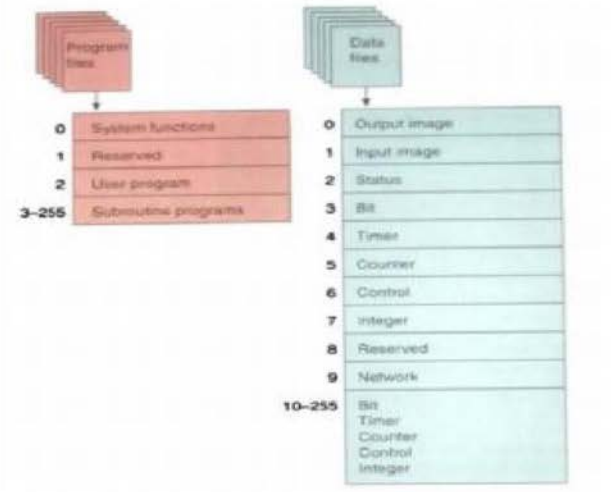


Figure: Program and data file memory organization for an Allen-Bradley SLC-500 controller

Data Files:-

The data file portion of the processor's memory stores input and output status, processor status, the status of various bits, and numerical data. All this information is accessed via the ladder logic program. These files are organized by the type of data they contain and may include:

- **Output (file 0) -** This file stores the state of the output terminals for the controller.
- **Input (file 1)**-This file stores the status of the input terminals for the controller.
- **Status (file 2)** - This file stores controller operation information. This file is useful for troubleshooting controller and program operation.
- **Bit (file 3)**-This file is used for internal relay logic storage.

- **Timer (file 4)**-This file stores the timer accumulated and preset values and status bits.
- **Counter (file 5)**-This file stores the counter accumulated and preset values and status bits.
- **Control (file 6)**-This file stores the length, pointer position, and status bit for specific instructions such as shift registers and sequencers.
- **Integer (file 7)**-This file is used to store numerical values or bit information.
- **Reserved (file 8)** - This file is not accessible to the user.
- **Network communications (file 9)** - This file is used for network communications if installed or used like files 10-255.
- **User-defined (files 10-255)**-These files are user-defined as bit, timer, counter, control, and/or integer data storage.

PROGRAM SCAN OF A PLC:-

- During each operating cycle, the processor reads all the inputs, takes these values, and energizes or de-energizes the outputs according to the user program. This process is known as a scan. Figure shown below shows a single PLC scan, which consists of the I/O scan and the program scan. Because the inputs can change at any time, the PLC must carry on this process continuously.

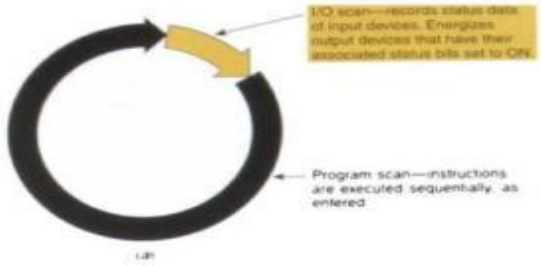


Figure: Single PLC scan

- The PLC scan time specification indicates how fast the controller can react to changes in inputs. Scan time varies with program content and length. The time required to make a single scan can vary from about 1 ms to 20 ms. If a controller has to react to an input signal that changes states twice during the scan time, it is possible that the PLC will never be able to detect this change.
- For example, if it takes 8 ms for the CPU to scan a program, and an input contact is opening and closing every 4 ms, the program may not respond to the contact changing state. The CPU will detect a change if it occurs during the update of the input image table file, but the CPU will not respond to every change. The scan is normally a continuous and sequential process of reading the status of inputs, evaluating the control logic, and updating

Figure shown below illustrates this process.

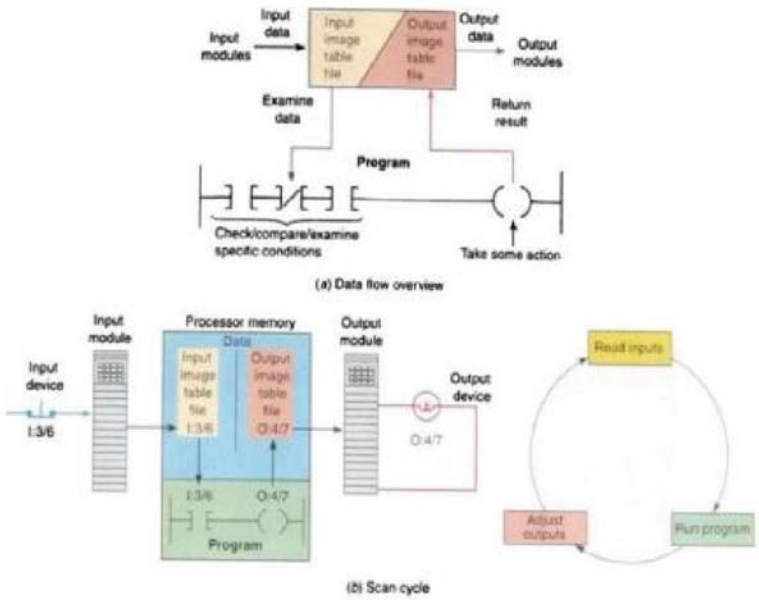


Figure: Scan Process

Figure: Scan Process

- When the input device connected to address I:3/6 is closed, the input module circuitry module circuitry senses a voltage and a 1 (ON) condition is entered into the input image table bit I:3/6. During the program scan, the processor examines bit I: 3/6 for a 1 (ON) condition.
- In this case, because input I: 3/6 is 1, the rung is said to be TRUE. The processor then sets the output image table bit O: 4/7 to 1. The processor turns on output O: 4/7 during the next I/O scan, and the output device (light) wired to this terminal becomes energized. This process is repeated as long as the processor is in the RUN mode. If the input device were to open, a 0 would be placed in the input image table. As a result, the rung would be called FALSE. The processor would then set the output image table bit O: 4/7 to 0, causing the output device to turn off.

CONTROL INSTRUCTIONS OF PLC:-

- Figure given below shows typical program control instructions.
- Instructions comprising the override instruction group include the master control reset (MCR) and jump (JMP) instructions.
- These operations are accomplished by using a series of conditional and unconditional branches and return instructions.

- Hardwired master control relays are used in relay circuitry to provide input/output power shutdown of an entire circuit. Figure shown below is a typical hardwired master control relay circuit. In this circuit, unless the master control relay coil is energized, there is no power flow to the load side of the MCR contacts. The master control relay circuit shown in Figure below could not be programmed into the PLC as it appears because it contains two vertical contacts. For this reason, most PLC manufacturers include some form of master control relay as part of their instruction set. These instructions function in a similar manner to the hardwired master control relay; that is, when the instruction is true, the circuit functions normally, and when the instruction is false, outputs are switched off. Because these instructions are not hardwired but programmed, for safety reasons they should not be used as a substitute for a hard· wired master control relay, which provides emergency I/O power shutdown

DIFFERENCE BETWEEN PLC AND PC:-

PLC:-

- The input- output capabilities are large and can be used in various industrial processes.
- The input-output cards are mounted on racks and are visible.
- The noise is more.
- The possibility of mishap exists as input-output cards are on the rack.
- The connections are rugged, accessible and organized.
- The processor architecture is simple.
- It is used for controlling industrial processes and control logic programming.
- The interface is by using simple devices such as indicators push buttons etc.
- It uses ladder programming.
- It has limited expandability.

PC:-

- Limited input-output capabilities interms of analog and digital cards which are suited for laboratory purposes.

- The input-output cards are mounted on board and are not visible.
- Noise is less
- No possibility of mishap
- There exists a bunch of cables and very small room available for cable connections.
- The processor architecture is complicated.
- It is used for many advanced computation and high level programming language is used.
- The easy and good interface is not possible.
- It uses microprocessor programming
- It has great flexibility and high reliability.

LADDER RUNG DIAGRAM:-

- The main function of the ladder logic diagram program is to control outputs based on input conditions. This control is accomplished through the use of what is referred to as a ladder rung. In general, a rung consists of a set of input conditions, represented by contact instructions, and an output instruction at the end of the rung, represented by the coil symbol (Fig. given below).

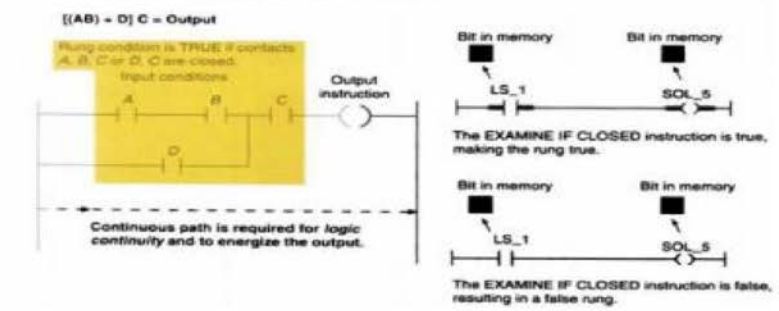


Figure: Ladder rung

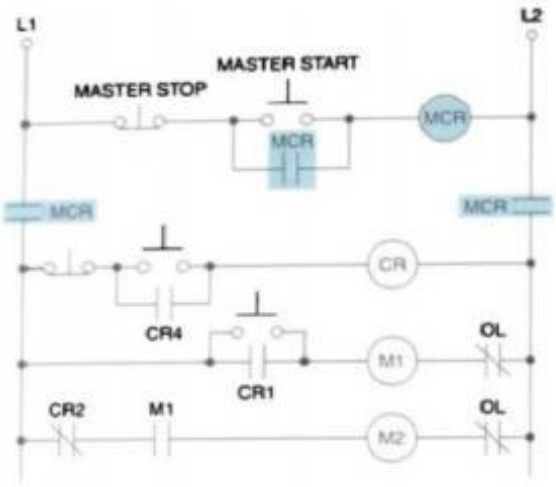


Figure: Hardwired master control relay circuit

- The master control reset instruction can be programmed to control an entire circuit or to control only selected rungs of a circuit. In the program of figure shown below, the MCR is programmed to control an entire circuit. When the MCR instruction is false, or de-energized, all non retentive (nonlatched) rungs below the MCR will be de-energized even if the programmed logic for each rung is true. All retentive rungs will remain in their last state. The MCR instruction establishes a zone in the user program in which all nonretentive outputs can be turned off simultaneously. Therefore, retentive instructions should not normally be placed within an MCR zone because the MCR zone maintains retentive instructions in the last active state when the instruction goes false.

- Each contact or coil symbol is referenced with an address number that identifies what is being evaluated and what is being controlled. The same contact instruction can be used throughout the program whenever that condition needs to be evaluated.
- For an output to be activated or energized, at least one left-to-right path of contacts must be closed. A complete closed path is referred to as having logic continuity.
- When logic continuity exists in at least one path, the rung condition is said to be TRUE. The rung condition is FALSE if no path has continuity.
- During controller operation, the processor determines the ON/OFF state of the bits in the data files, evaluates the rung logic, and changes the state of the outputs according to the logical continuity of rungs. More specifically, input instructions set up the conditions under which the processor will make an output instruction true or false.
- These conditions are as follows:
    1. When the processor finds a continuous path of true input instructions in a rung, the

    2. When the processor does not find a continuous path of true input instructions in a rung, the OTE input instruction will become (or remain) false. We then say that rung conditions are FALSE.

TIMER:-

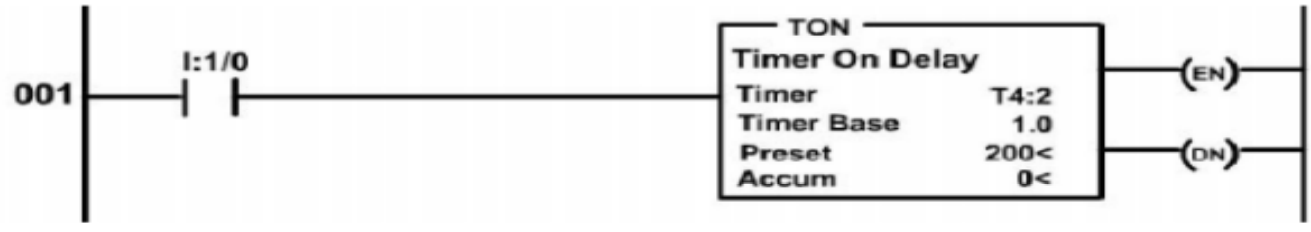
All PLC's have timer instructions. Timers are output instructions that are internal to the programmable logic controller. Timers provide timed control of the devices that they activate or de-activate.

Basic functions of timer

- Timers are used to delay an action.
- Timers are used to run an operation for a predetermined period of time.
- Timers are also used to record the total accumulated time of continuous or intermediate events.

Timer's Instructions:-

Timers consists of following parts: timer address, preset value, timer base, and accumulated value, as shown in figure below.



In the above figure, term instruction name is timer on delay (TON), timer base is 1.0 seconds, timer address is T4:0, accumulated value of zero (0) and a preset value of 200. Each timer instruction has three very useful status bits. These bits are timer enable (E N), timer timing (TT), and timer done (DN). There are 3 types of timers: On- delay timer, Off-delay timer, and retentive timer.

On delay timer:-

- Use this instruction to program a time delay after instructions become true.
- On – delay timers are used when an action is to begin a specified time after the input becomes true. For example, a certain step in the manufacturing is to begin 45 seconds after a signal is received from a limit switch. The 45- seconds delay is the on-delay timers preset value.

Off- delay timer:-

- Off- delay timer instructions is used to program a time delay to begin after rung input goes false.
- As an example, when an external cooling fan on a motor is provided, the fan has to run all the time the motor is running and also for certain time (say 10min) after the motor is turned off. This is a ten minute off- delay timer. The ten-minute timing period begins as soon as the motor is turned off.

Retentive timer:-

- Retentive timer is a timer which retains the accumulated value in case of power loss, change of processor mode or rung state going from true to false (rung state transition).
- Retentive timer can be used to track the running time of a motor for its maintenance purpose. Each time the motor is turned off, the timer will remember the motor's elapsed running time. The next time the motor is turned on, the time will increase from there. This timer can be reset by using a reset instruction.

Reset:-

- This instruction is used to reset the accumulated value of counter or timer.
- It is used to reset a retentive timer's accumulated value to zero.

A typical timer element:-

A timer element is made up of three 16 bit words:

- Word 0: 3 status bits (EN, TT, DN).
- Word 1: Preset values.
- Word 2: Accumulated value.

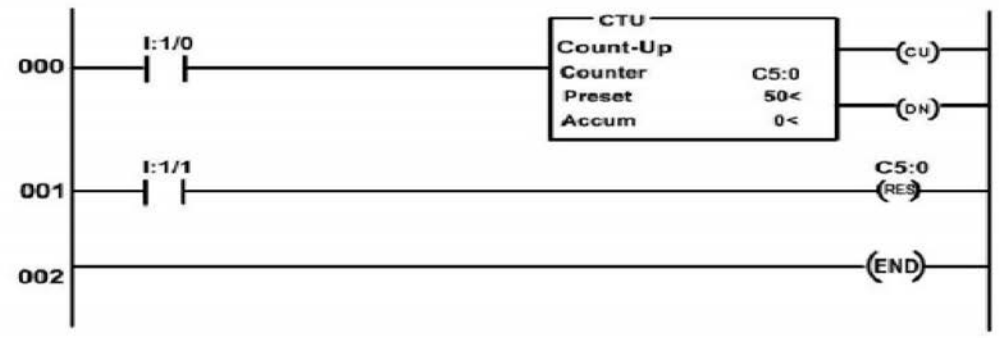
EN	TT	DN	Reserved Bits
Preset Value			
Accumulated Value			

COUNTER:-

A counter is a simple device intended to do one simple thing-count. Every PLC has counter instructions. Using counters sometimes be little challenging because many manufacturers seem to use them different way. In other words, the instruction symbol used and method of programming will change for different manufacturers.

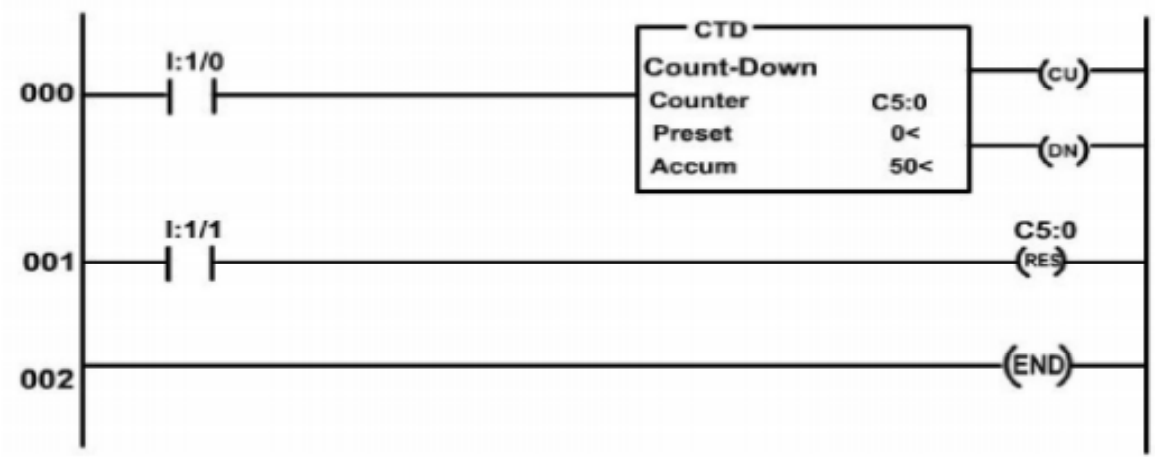
A typical counter counts from 0 up to a predetermined value, called the "preset" value. For example, if you wanted to count from certain value, say from 0 to 50, you would be counting up using a count-up or up-counter. Here, the number "50" is the predetermined value, which is nothing but preset value. The current count or accumulated count is called as the "accumulated value". If our counter had counted 25 pieces that had passed on the conveyor, the accumulated count would be 25. When all 50 pieces had passed on the conveyor, the preset value and counter accumulated value would be equal. At this point the counter would signal other logic within the PLC program that the batch of 50 was completed and it should now take some action. The next action the PLC has to take is to move the box containing 50 parts on to the next station for carton sealing. To start counting the next batch, a reset instruction would be used to reset the counter's accumulated value back to zero.

The fig below shows an up-counter, counting from 0 to 50.



The fig below shows a down-counter, counting from 50 to 0.

The fig below shows a down-counter, counting from 50 to 0.



PLC Counter Instructions

Instruction	This instruction is used to	Functional description
Count Down	Count down from a desired value to zero	An operator interface display shows the operator the number of parts remaining to be made for a lot of,say 50 parts ordered.
Count Up	Count from zero up to a desired value	Counting the number of parts produced during a specific work shift or batch.Also counting the number of rejects from a batch.
High-speed Counter	Count input pulses that are too fast,from normal input points and modules	Most fixed programmable logic controllers have a high-speed set of input points that allow interface to high-speed inputs. Signals from an incremental encoder would be a typical high-speed input.
Counter Reset	To reset a counter or timer	Used to reset a counter to zero so that another counting sequence can begin.

As for explanation, let us take ALLEN-BRADELY counters:

In Allen-bradely counter, the default counter file is file 5. The counter data is stored in counter file. Each counter consists of three 16-bit words and is known as "counter element". In a single processor file, there can be many counter files. Any data file, which is greater than file 8 can be assigned as an additional counter file. Each counter file can have up to 256 counter elements. A counter instruction is one element. A counter element is made up of three 16-bit words. Thus, the counter instruction contains the three parts i.e. word0, word1 and word2.

- Word zero is for status bits. Status bits include CU, CD, DN, OV, UN, and UA. Along with their associated instructions.

Addressing a counter

1) To address the counter as a unit the addressing format used is C5:4.
Where, C= C identifies this as a counter file.
5= This is counter file 4 which is default. Any unused file from 10 to 255 can be assigned to counters.
:4= The colon used here is called the file separator. It separates the file, file 5, from the specific counter, in this case, counter 4 in counter file 5.
2) To address the counter 14's accumulated value, the address used is C5:14.ACC.
Where, C= C identifies this as a counter file.
5= This represents the counter file 5.
:14= The colon is called the file separator. The colon separates the file, file 5, from the specific counter; in this case, counter 14 in counter file 5.
. = The point is called the word delimiter. The word delimiter is used to separate the counter number, called the structure, from the sub element. The sub element is ACC for the accumulated value. Similarly, preset value can be accessed as C5:14.PRE.

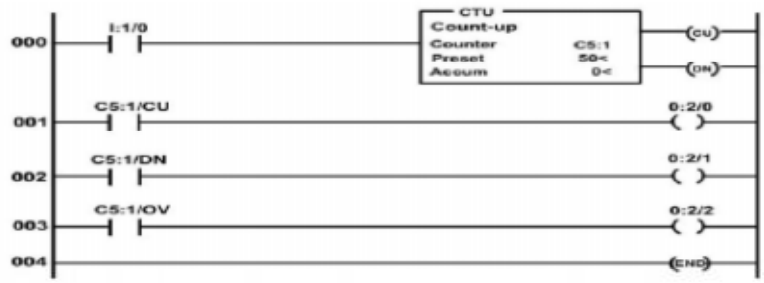
The addressing format of counter status bits is as follows:

- C5:14/DN is the address for counter file 5, counter 14's done bit.
- C5:14/CU is the address for counter file 5, counter 14's count-up-enable bit.
- C5:14/EN is the address for counter file 5, counter 14's enable bit.

Working of a counter:-

- A counter instruction is always an output instruction. The counter instruction counts each time the input logic changes the rung state from false to true. This input logic can be signal from an external device, for e.g. limit switch or sensor, or a signal from internal logic. Every time the counter instruction sees a false-to-true rung transition, a count-up counters accumulated value is incremented by one.
- The working of down-counter is little different. Each time when count-down counter sees a false-to-true rung transition, its accumulated value is decremented by one. Since the accumulated value gets decremented by 1 when each time the input logic changes the rung state from false to true, the accumulated value must be chosen as the starting point of the count.
- Counters are retentive in nature. The counter will retain its accumulated value or the on or off status of the done, overflow and underflow bits through a power loss.

The count-up instruction



The count-up instruction is used if we want a counter to increment one decimal value each time it register a rung transition from false to true. Each time input I:1/0 has a transition from off to on, counter C5:1 will increment its accumulated value by one decimal value. The count-up-enabled bit, on rung 001 is set when the rung conditions are true, or enabled. In rung 002, the done bit, DN, is set when the accumulated value is equal to or greater than the preset value. In the event of wrap from +32,767 to -32,768, the accumulated value becomes less than the preset value and the done bit will not be reset. In rung 003, the count-up overflow bit, OV, is set whenever the count-up counters accumulated value wraps from +32,767 to -32,768.

The count-down instruction

This instruction is used when we want to count down over the range of +32,767 to -32,768. Accumulated value will be decremented by one count, each time the instruction sees a false-to-true transition. Count-down instruction has many applications, for example: if we want to display the remaining number of parts to be filled for a specific order say 50 parts, then, a count-down instruction can be used. In this example, the accumulated value will be set as 50 and the preset value will be 0. Each time a part is completed and passes the sensor, the accumulated value will be decremented by one decimal value, as shown in the figure below.