BHUBANANANDA ODISHA SCHOOL OF ENGINEERING

CUTTACK ODISHA

APPLIED ELECTRONICS AND INSTRUMENTATION ENGINERING

PLC LAB MANUAL

PREPARED BY:

AE&I DEPARTMENT

PRAFULLA KUMAR PANDA

SUBMITTED TO

AE&I DEPARTMENT

Practical: 4 Periods per weak **Sessional:-**

Total periods: - **Term End Exam: -**

Examination: -3 hours **Total mark: -**

SL NO.	NAME OF THE EXPERIMENTS	PAGE NO.
1	To study hardware and software associated with PLC.	
2	To understand Simple Ladder program.	
3	To develop a ladder program for DOL starter.	
4	To develop an application using On-Delay timer.	
5	To develop an application using Off-Delay timer	
6	To Develop an application using UP/DOWN counter	
7	To Study computational / arithmetic instructions used in PLC ladder programming	
8	To understand working of PID function block	
9	PLC Program to Control Traffic Lights	

VISSION OF THE DEPARTMENT

To produce efficient professional in applied electronics and instrumentation engineering and other allied area and with update technical knowledge to meet the challenge of society in relevant sector.

MISSION OF THE DEPARTMENT

- **M1** To produce the student competent in applied electronics and instrumentation engineering with social, environmental and human through quality education training.
- **M2** Provide knowledge of basic science, applied mathematics instrumentation technology and communicative skills to identify and solve problems related to applied electronics and instrumentation engineering.
- **M3** To enable the students to acquire various parameter measurement and automatic control technology used for industrial automation and include quality of leadership teamwork in collaboration with parents, alumni and industries.

POGRAM EDUCATIONAL OBJECTIVE

- **PEO1** To provide Student with a solid fundamental foundation in basic science, electronics instrumentation and interdisciplinary subjects that is necessary excel in professional career, higher education.
- **PEO2** To Prepare student to meet the needs and face the challenges of real life as well as industry automation and digitalization in terms of technical economic and social Feasibility.
- **PEO3** To inoculate professionalism communication skills, attitudes teamwork and to adapt to the current trends by engaging in lifelong learning.
- **PEO4** To utilize the technology of applied Electronics and instrumentation in all relevant fields as per need.

Experment-1

Aim: To study hardware and software associated with PLC.

Objective:

1. Learn the basics and hardware components of PLC
2. Understand configuration of PLC system
3. Study various building blocks of PLC

Evolution of PLC: -

When the first electronic machine control was designed, relays were to control the machine logic. Relay logic has its own limitations.

1. Less reliability
2. The delay involved in switching of contacts
3. Less flexibility and difficult troubleshooting due to hard wired connection

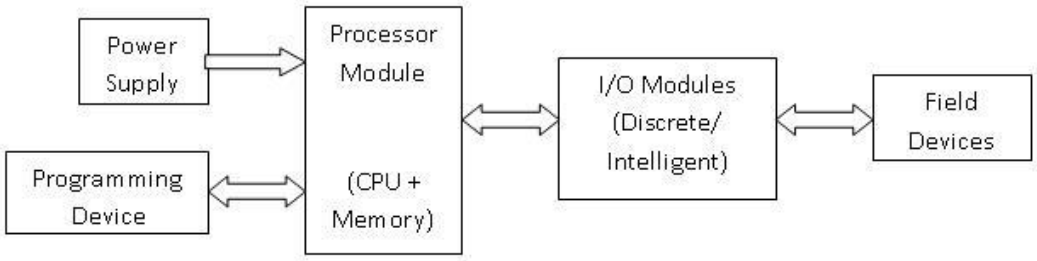
What is PLC?

A Programmable Logic Controller , PLC, or Programmable Controller is an electronic deviceused for Automation of industrial processes, such as control of machinery on factory assembly lines. A programmable controller is a digitally operating electronic apparatus which uses a programmable memory for the internal storage of instructions for implementing specific functions, such as logic, sequencing, timing, counting and arithmetic, to control various machines or processes through digital or analog input/output devices.Unlike general purpose computers, the PLC is designed for multiple inputs and output arrangements, extended temperature ranges, immunity to electrical noise, and resistance to vibrations and impacts.

Programs to control machine operation are typically stored in battery-backed or non volatile memory. A PLC is an example of a real time system since output results are produced in response to input conditions within a bounded time, otherwise unintended operation results.

Basic Components of PLC: -

1. CPU and Memory module
2. Power supply
3. Input and output module
4. Programming device



- CPU and Memory Module: -

  This is the device where PLC program is stored and processed. The size and type of CPU determines the programming functions available, size of the application logic available, amount of memory supported, and processing speed.

- Power Supply: -

  The power supply provides power for the PLC system. It provides internal DC current to operate the processor logic circuitry and input/output assemblies. This can be built into the PLC or an external unit. Common voltage levels required by the PLC are 24Vdc, 120Vac, 220Vac. , is used to determine temperature.

- Input and Output Module: -

  Inputs carry signals from the field (process) to the controller. Various types of inputs can be switches, pressure sensors, transmitters etc. The field devices to whom PLC sends the results of logical operations are the output devices. These are the actuators that adjusts or control the process, motors, lights, relays, pumps, etc. Many types of inputs and outputs can be connected to a PLC and they can be categorized mainly as analog and digital. Digital inputs and outputs operate on discrete or binary change i.e. on/off, open/close. Analog inputs and outputs change continuously with reference to time.

- Programming Device: -

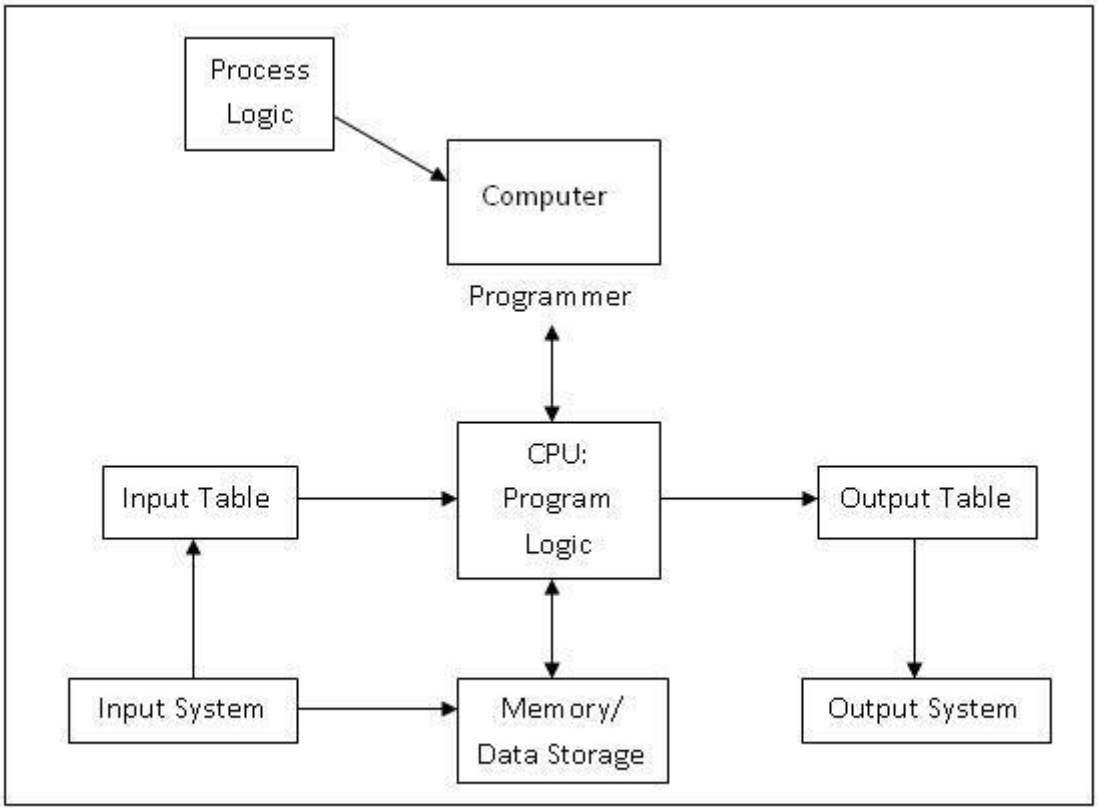
- The PLC is programmed using a special software using computer or hand-Held Terminal (HHT) that can load and change the logic inside.

Operation of a PLC system: -:

The operation of the PLC is determined by 3 steps.

1. Reading the field status form input devices

2. Execution or solving the logic, and
3. Updating the output devices status.

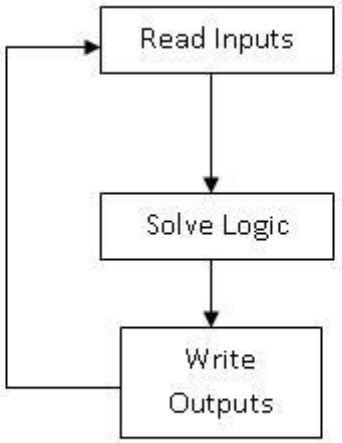


PLC Program: -

PLC Program is a Logic that is executed by the CPU. This logic can be written in the form of Ladder diagram, Instruction List, Sequential Function Charts, Structured text or Functional block diagram. These are the languages used for writing logic as per IEC standard. The program is then downloaded to the PLC. This is usually done by temporarily connecting the PC or HHT to the PLC. Once the program is downloaded to the CPU, it is usually not necessary for the PC to remain connected.

PLC Scan: -

Once the program is downloaded in the CPU, the PLC is switched to "run" mode and the PLC executes the application program. The CPU regularly reads the status of the input devices, and sends data to the output devices as per the logical results after execution of the program. The process of Initialization when power is turned on, reading inputs, executing logic, and modifying outputs is called as PLC Scan Cycle.



Memory

The logic or application program is stored in memory. As the PLC executes logic, it may also read and store values to memory. The values may be referenced by the application program.

Input and Output Devices

Two major types of Input/Output modules are

1. Digital - binary devices which must be in one of the two states: on or off.
2. Analog - continuous devices - sense and respond to a range of values.

- Digital I/O

  Common digital field input devices include pushbuttons, limit switches, photo sensor etc. Common digital output devices include relays, motor starters, and solenoid valves.

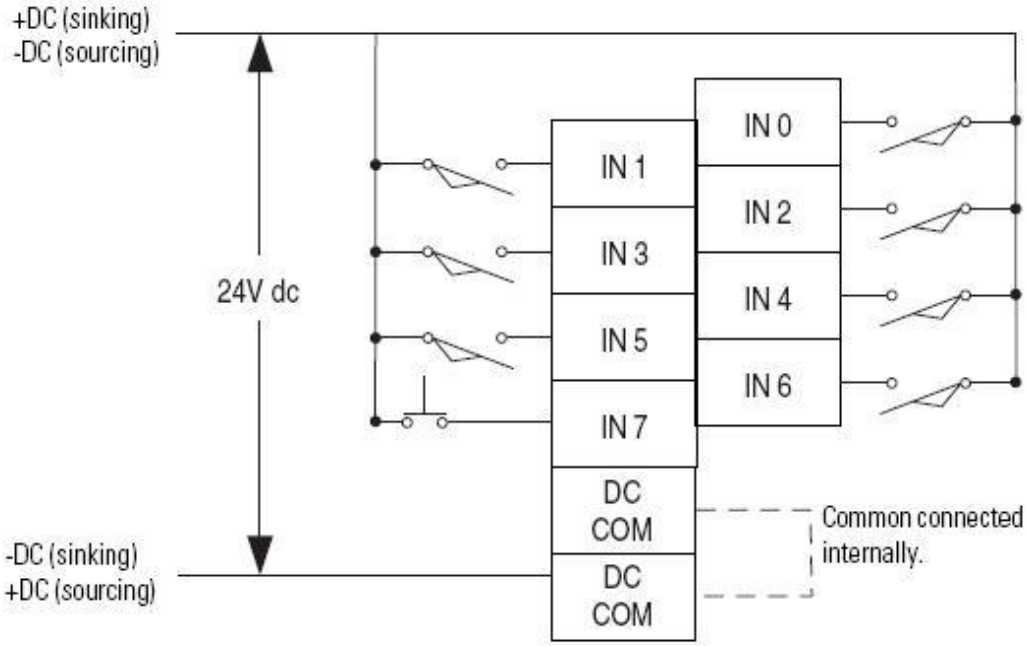
- Analog I/O

  Common analog input devices are transmitters used for sensing various parameters. Common output signals include motor speed, valve position, air pressure, etc.

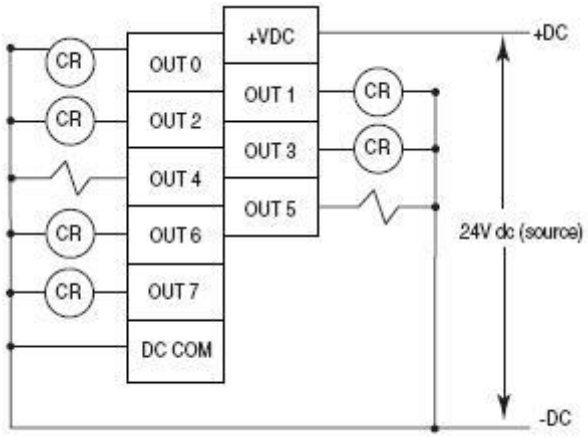
I/O modules connect "real world" field devices to the controller. They convert the electrical signals used in the field devices into electronic signals that can be used by the control system, and translate real world values to IO table values.

I/O Wiring: -

Example of Input Module Wiring Diagram: -



Example of Output Wiring Diagram: -



Sinking and Sourcing Operation: -

- If the device provides current to during it's on state, the device is said to be souring current.
- If the device receives current in the ON or true state, then it is sinking current.

Procedure

1. Once you open the simulator, at the left side you will see the library of nomenclature and symbols.
2. You have to classify the symbols in the various categories as AI, AO,DI and DO.
3. Left click on the nomenclature and symbols one by one and drag it to the right side (Workspace) to build the architecture of the PLC.

Experiment-2

Aim : To understand Simple Ladder program.

Objective:-

1. Develop a ladder using standard procedure.
2. Solve the problem using ladder programming.

Pre-requisites:-

Basics of Digital Electronics and Boolean Algebra:

Digitization is a process where continuous analog signal is converted into a finite number of discrete states. These states are well separated so that noise does not create errors.

The resulting digital signal has following advantages:

1. storage over arbitrary periods of time.
2. flawless retrieval and reproduction of the stored information.
3. flawless transmission of the information.

Some information is essentially digital. Hence it is natural to process and manipulate such information using purely digital techniques. Examples are numbers and words. The drawback to digitization is that a single analog signal (e.g. a voltage which is a function of time, like a stereo signal) needs many discrete states, or bits, in order to give a satisfactory reproduction. ', 'l'

Boolean Logic	Boolean Algebra	Voltage State (positive true)	Voltage State (negative true)
True (T)	1	High(H)	Low(L)
False (F)	0	Low(L)	High(H)

Logic

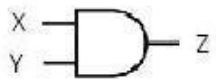
What can a digital circuit do? The simplest task we can think of is a combinational type of logic decision. For example, we can design a digital electronic circuit to make an instant decision based on some information. Here we emphasize “instant” in the decision making process. That means, the process has no time delay. , is used to determine temperature.

X = It is a sunny day? Yes
Y = Is it Sunday or holiday? Yes
Action Z = Go for shopping
The rule is Z = X and Y. The circuit is a simple AND gate.

Truth Tables:

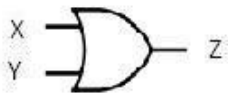
X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

Truth Table of AND



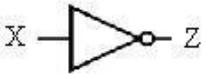
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

Truth Table of OR



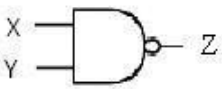
X	Z
0	1
1	0

Truth Table Of NOT



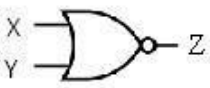
X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

Truth Table Of NAND



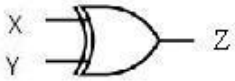
X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

Truth Table Of NOR



X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

Truth Table Of XOR (Exclusive OR)



Boolean Algebra:

Logic can also be expressed in algebraic form. e.g.Truth Table for AND gate:

X	Y	Z
0	0	0
1	0	0
1	1	1

Boolean Algebra Simplification:-

Basic Laws:

Commutative:	$X+Y = Y+ X$ $X.Y = Y.X$
Associative:	$X+Y+Z = (X+Y) + Z = X+(Y+Z)$ $X.Y.Z = (X.Y).Z = X. (Y.Z)$
Distributive:	$X.(Y+Z) = X.Y + X.Z$
De Morgan's:	$\overline{X+Y} = \bar{X} . \bar{Y}$ $\overline{X . Y} = \bar{X} + \bar{Y}$

Theory

Introduction

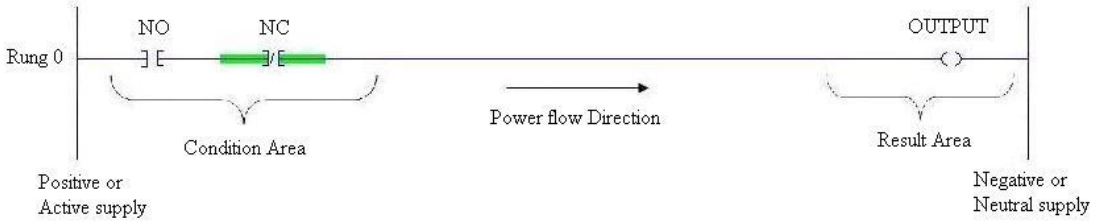
Each manufacturer of PLC systems has own style of writing the instructions. Different PLCs has different instruction sets but even some common basic instructions are shared by all the PLCs. All manufacturers give different software packages for programming PLCs. Ladder is most commonly used programming language. Prior to PLCs, relay logic was used in industry. Ladders were developed to mimic or imitate relay logic.

Relay Logic / Instructions A relay is simple magnetic device which acts as a control switch. When the switch is on, current will flow through the coil on iron piece. This iron core acts a electromagnet and due to the magnetic field upper contact gets attracted towards lower one and circuit gets completed, allowing current to flow from load.

Ladder Programming

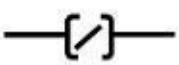
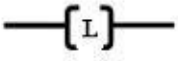
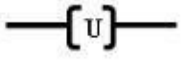
Ladder diagram is popular language of programming the PLCs. Ladder diagram shows the sequence of the logic execution which is presented diagrammatically. In ladder diagram, There are two vertical lines generally called as Phase (positive) or neutral. Rungs which show current flow in horizontal direction are the sequence in which the logic executes. The Analogous to relay, ladder has two main symbols which are contacts and output coil. Generally each rung has inputs (contacts) on left hand side and outputs (coil) on the right

hand side. These contacts and coils are called as bits of the relays. Each input and output are individual bit in I/O files. An instruction in ladder instructs PLCs how to respond to the bits in I/O files which are stored in the memory. Input contacts are the condition area, the conditions must be fulfilled to change the status of the output coils.



Typical Ladder Diagram

Most commonly used relay instructions used in PLC programming are as shown in the table below.

Instruction	Symbol	Description
Examine ON (Normally Open)		An input condition that is open when de-energized
Examine OFF (Normally Closed)		An input condition that is close when de-energized
Output coil		An output instruction that is true when the input conditions become true
Negated Output		An output instruction that is true all the time except when all input conditions true
Latch Output Coil		To hold an output ON
Unlatch		To unlatch the latched ON output

Procedure

Important steps for developing Ladder using Simulator:

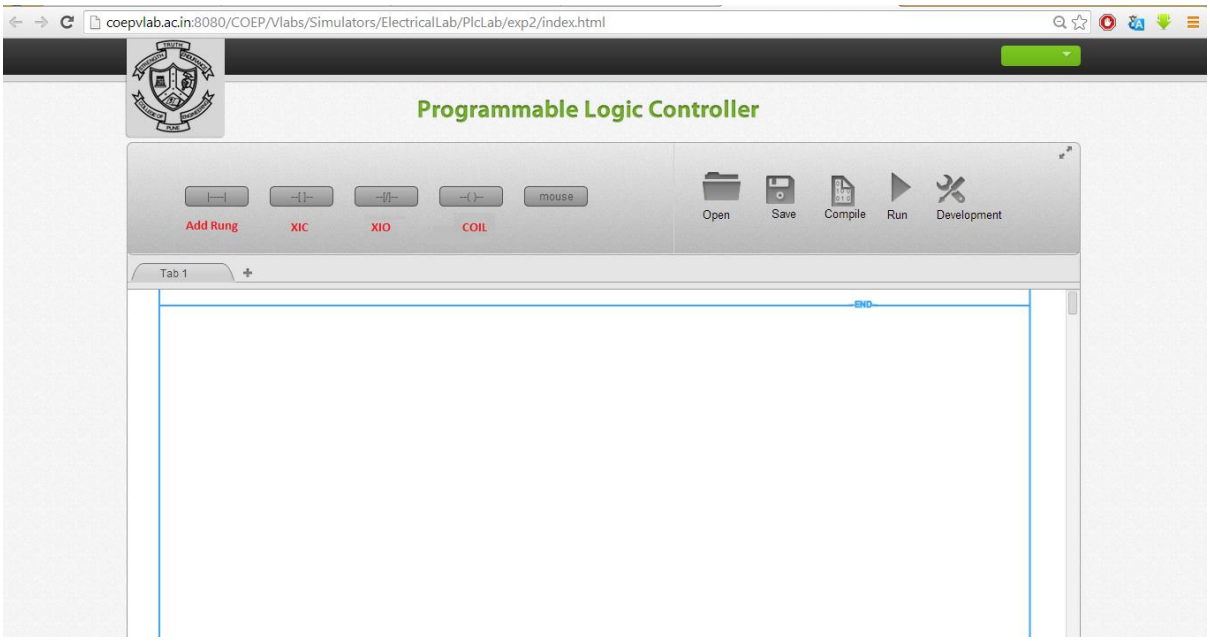
Please follow the steps so as to understand the procedure for developing ladder logic for various logical gates.

1. Prior to starting of the development of ladder diagram following steps needs to be understood:

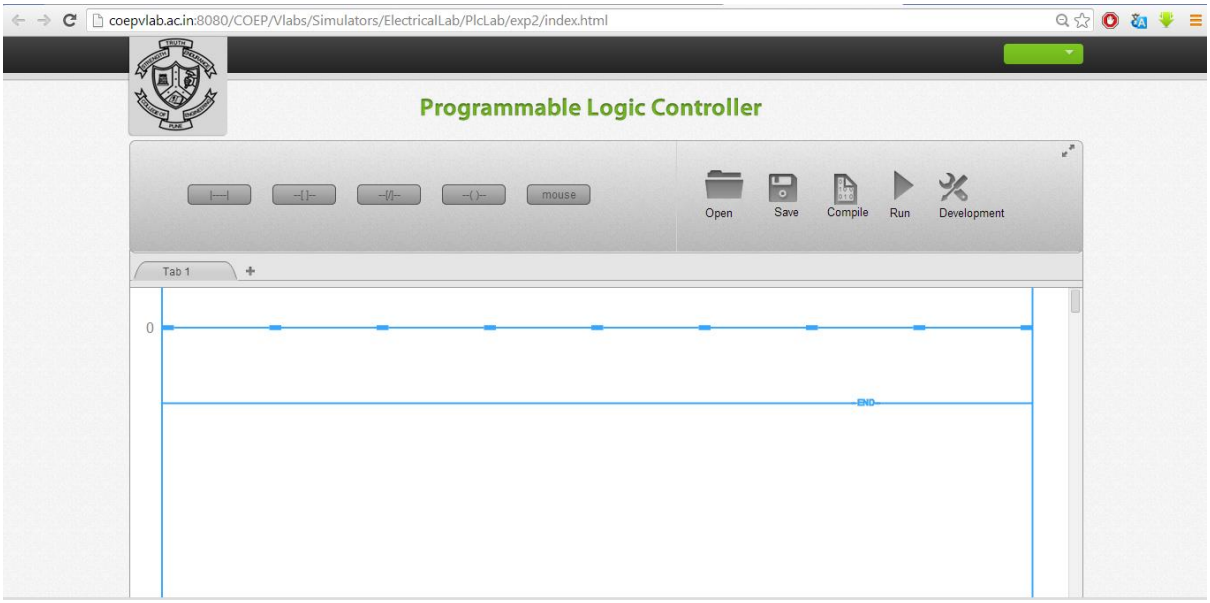
- Understand the problem statement like test the logic for OR gate, AND Gate etc.
- Develop the logic on paper and validate the logic by considering various cases
- Prepare the truth table and test the logic using all valid cases
- Go to simulator icon and click on the “Simulator” button
- The PLC simulator will be opened in new window.

2. The procedure for writing the ladder diagram in the work space is as follows:

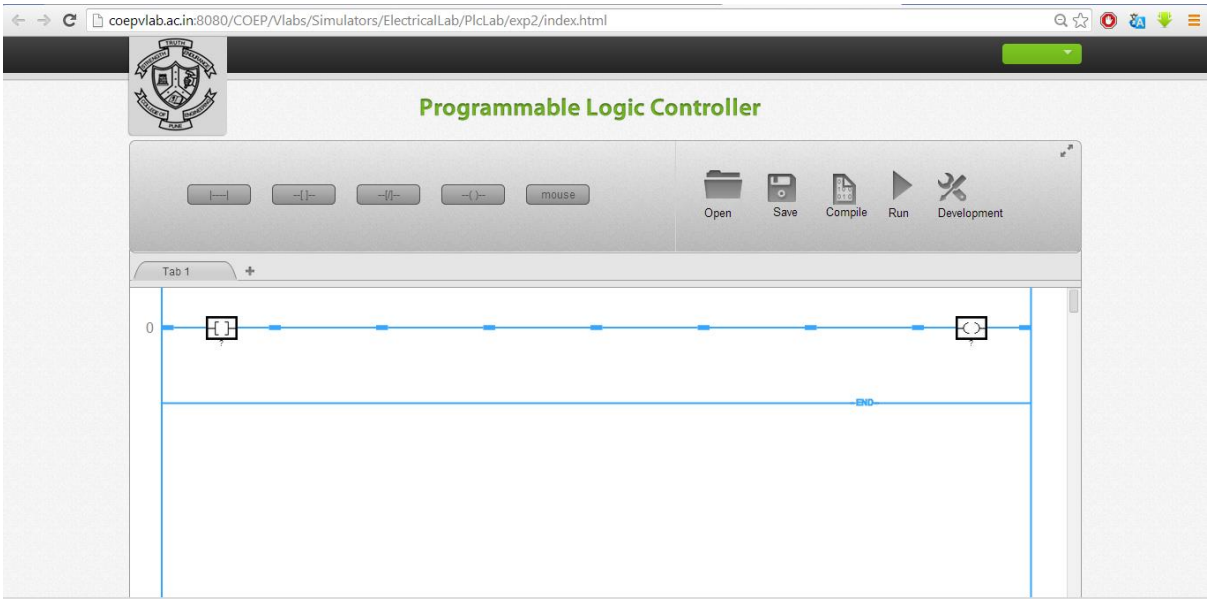
- The screen shot for the first window will appear like this.



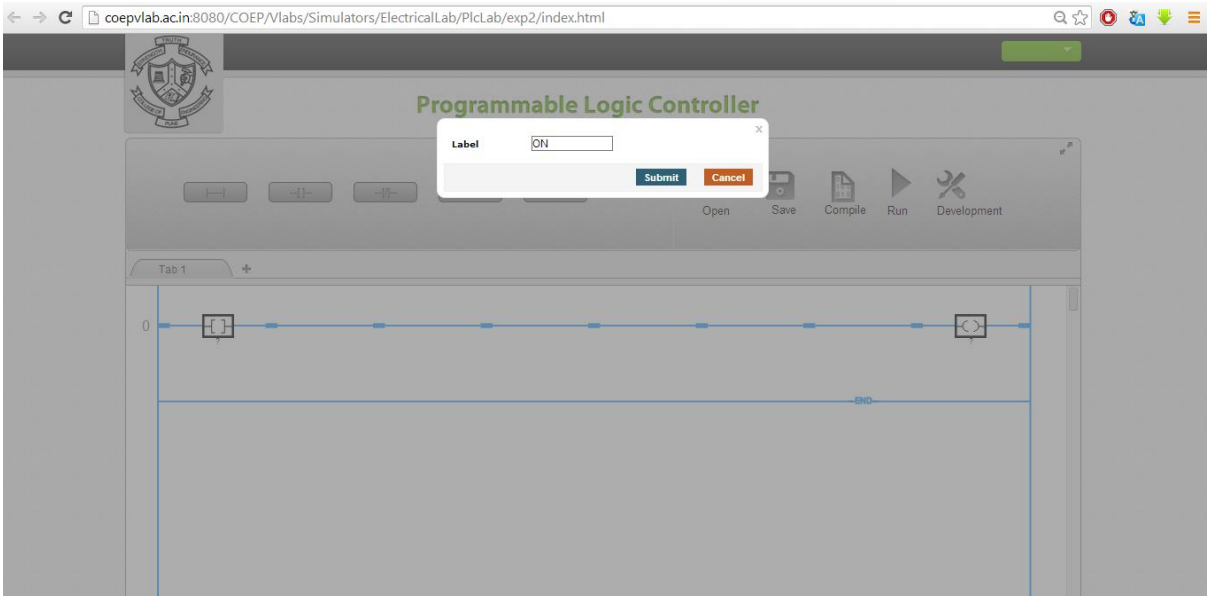
- Add a new rung by clicking on the 'Add Rung' icon. The window will appear like this:



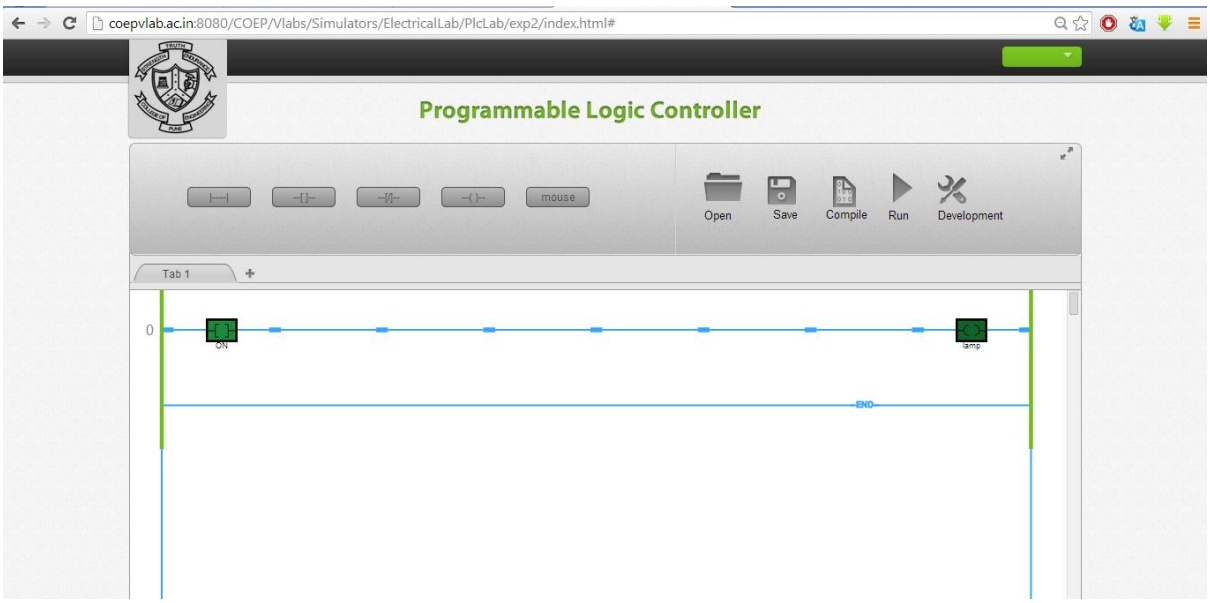
- Place the contacts as per the requirement by left clicking the appropriate contact shown at the top side. In the example demonstrated below one normally contact and one coil is placed as shown the figure.



- Right click on the contact or coil and you can give tag name like “start”, “stop” etc. Please ensure that the tag numbers are true replica of process connections. Similarly give tag name to coil like “motor”, “Lamp” etc. The final ladder will look like this.

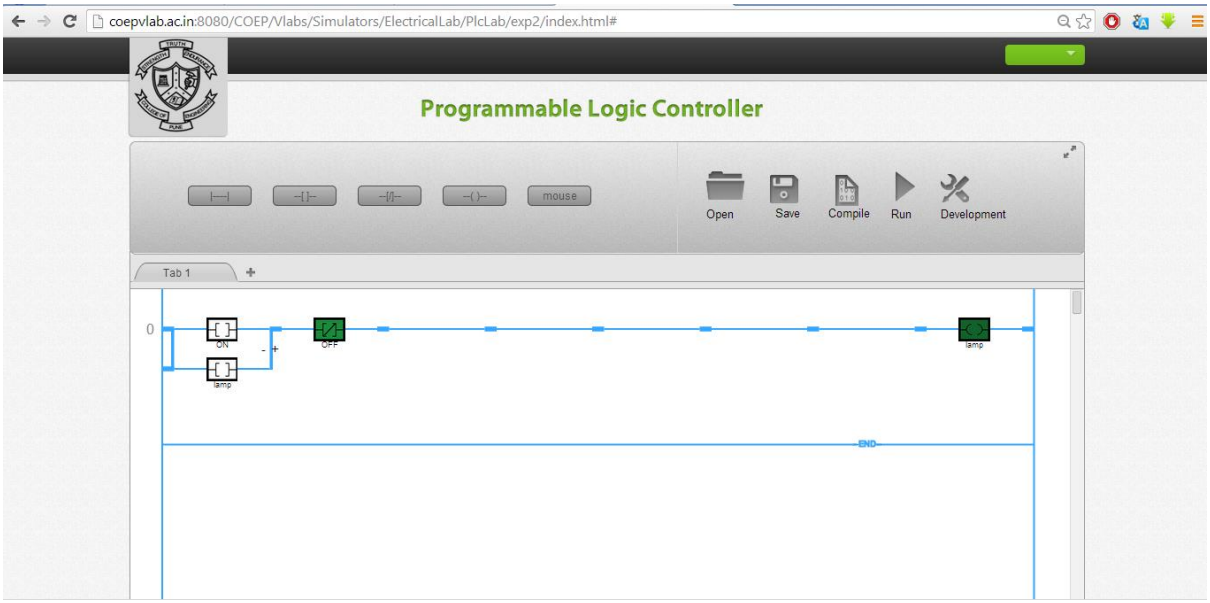


- Please note that the tag names are case sensitive and if you are using them in circuit as bit make sure that the correct tag name appears.
- Click the Compile button so the ladder will be ready for running. For testing the logic you need to click on Run. Both sides of the rung will become green and this is the indication of run mode. Please not that in run mode you can't make changes in the ladder. For modifications user has to go to development mode.
- By right clicking on the contact toggle the state of contact.Check the output contact status.



- Please remember the ladder contacts or the state of the inputs and outputs are always in de-energised state. The de-energised is that state wherein the contacts are in non-active state.
- You can once again toggle the contact and the output state will change. To add any contact you will have to go to development mode. Click on the rung and add contacts.

- To delete any contact or output right click on the contact and press “delete”, the contact will be deleted.
- You can add seven elements in series and 5 elements in parallel.
- To add element in parallel click on the node near contact where you wish to add parallel branch. Select the branch and click on the '+' sign to complete the loop. The screen will appear as shown below.



- Repeat the procedure and verify your logic.
- Similarly you can check the logic for OR, NOR, and NAND gates. Validate the truth tables and confirm the results.

Experiment-3

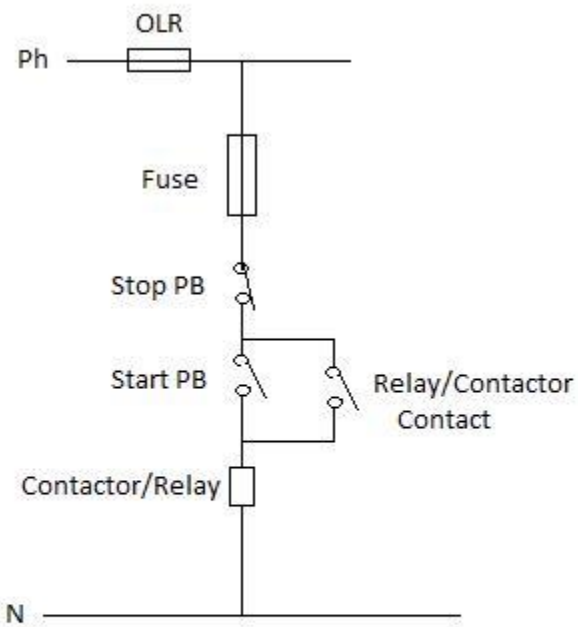
Aim : To develop a ladder program for DOL starter.

Objective:-

1. To understand working of DOL starter
2. Develop a ladder program for starting an electrical motor using DOL starter

Working of Direct-On-Line (DOL) starter:

One method of starting electric motors is using direct on line (DOL) or across the line starter. In this method full line voltage is applied to the motor terminals. This is simplest type of motor starter. An electrical wiring diagram for single phase DOL starter is shown below.

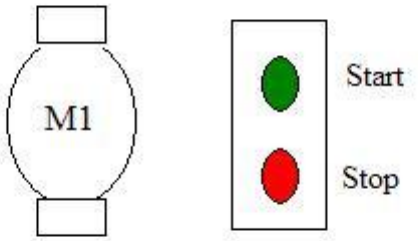


Control Circuit for DOL Starter

A DOL motor starter contains fuse and over load relay (OLR) for protection purpose. The starter can be contain momentary contact or maintained contact push buttons. The example considered here is momentary contact push buttons. For starting purpose normally open (NO) push button is preferred whereas normally closed (NC) push button is used to stop the motor. The excessive supply voltage drop causing high inrush current is the criteria to limit the use of DOL starter. Conveyor motors, water pumps are the applications where DOL starters are used.

Procedure

Problem Statement: To start a motor using DOL starter . The simple P&I ; diagram for this problem is as below.



Listing of Input and Output devices: Inputs: PB1- To start the motor PB2- To stop the motor Output: M1- Motor

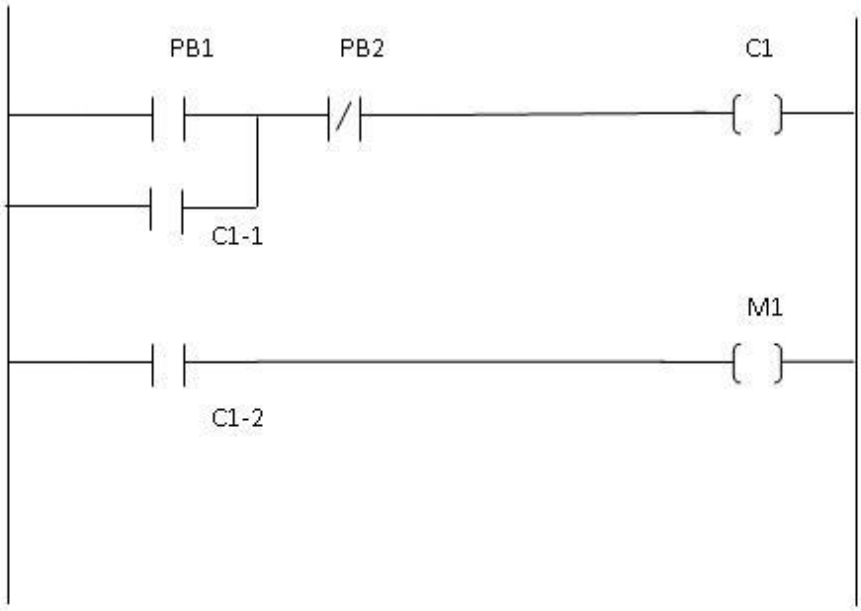
Sequence of Events :

1. When Start push button (PB1) is pressed, Motor (M1) has to start.
2. If Start pushbutton (PB1) is released and Stop pushbutton (PB2) is not pressed, Motor (M1) should remain on.
3. When Stop push button (PB2 is pressed, Motor (M1) has tol stop.
4. If stop push button is released and start is not pressed (released) motor shouldl remain off.

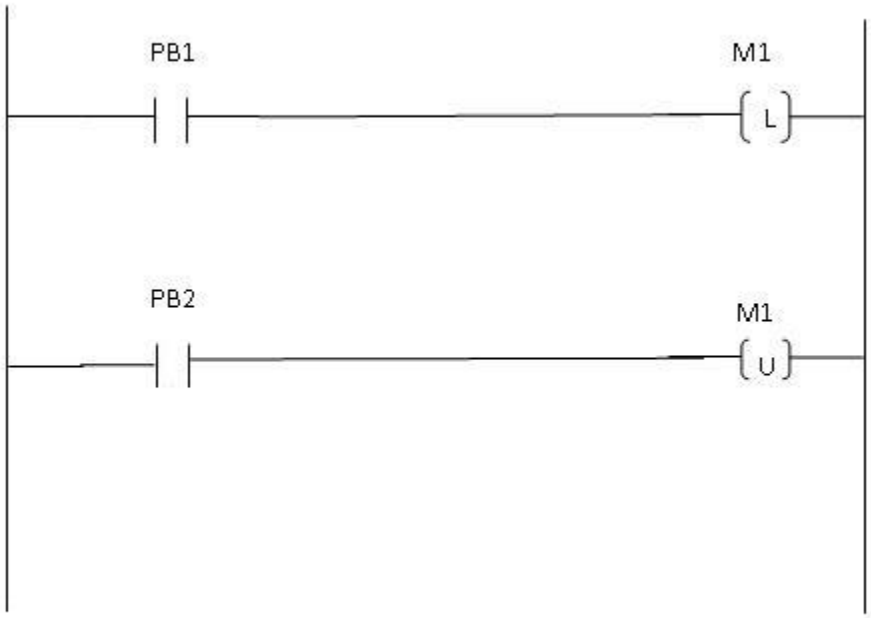
The Boolean equation to represent this sequence is

$$M1 = PB1.\overline{PB2}$$

The ladder diagram to implement these equations is shown below.



As the momentary contact push buttons are used here, the condition of PB1 is maintained through contact of coil C1. This contact is called as latching contact. The same sequence of event can be executed by using latch and unlatch instruction in the following way.

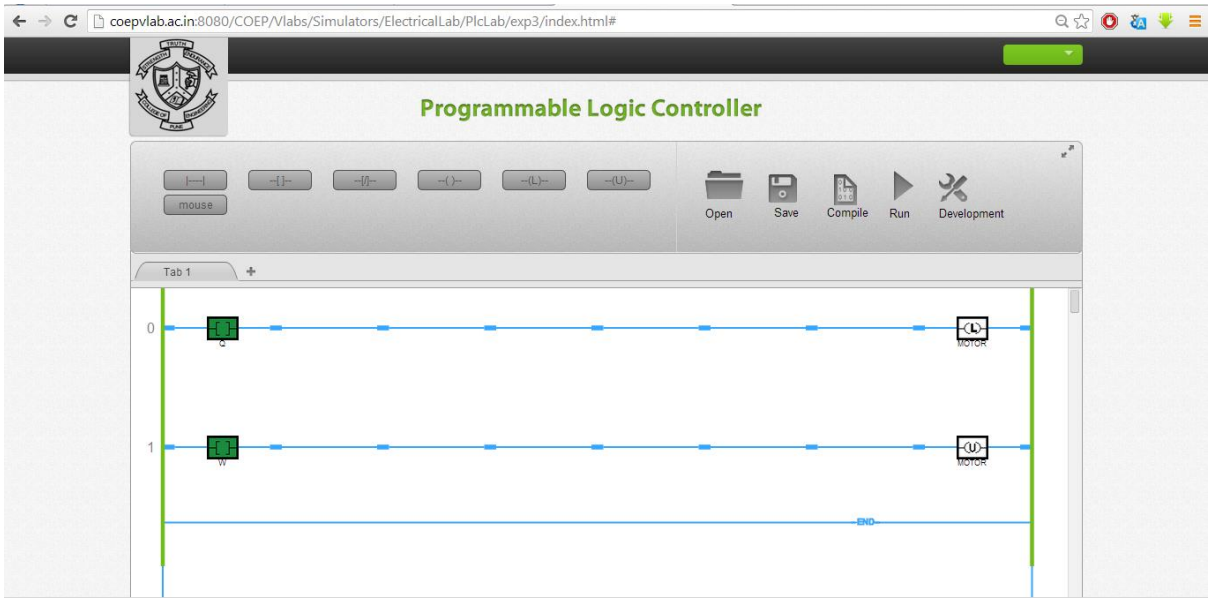


Procedure

Open the Simulator window as described in the last experiment

1. The Latch and unlatch instructions are used for holding the output status.
2. The tag name of latch and unlatch output bit must be same.
3. Once you toggle the input bit for the latch; even if you release it by toggling once again, the output bit remains latched.
4. To unlatch the output you will have to toggle the input bit in the unlatch rung and the output will be de-energised.

Execute the following ladder on simulator and observe the output status:



- You can develop ladder for a DOL "Direct On Line" starter using these instructions.
- You can also develop the logic using start and stop push buttons as explained under theory tab.
- Observe the output status at different input conditions.

Experiment-4

Aim : To develop an application using On-Delay timer.

Objective:

1. Study the timing diagram of On Delay Timer
2. Solve the assignment of Ton timer

PLC timers instruction is used to activate or deactivate a device after a preset interval of time.

Types of Timers available are:

1. On-Delay timer (TON)
2. Off-Delay timer (TOF)
3. Retentive timer on (RTO)

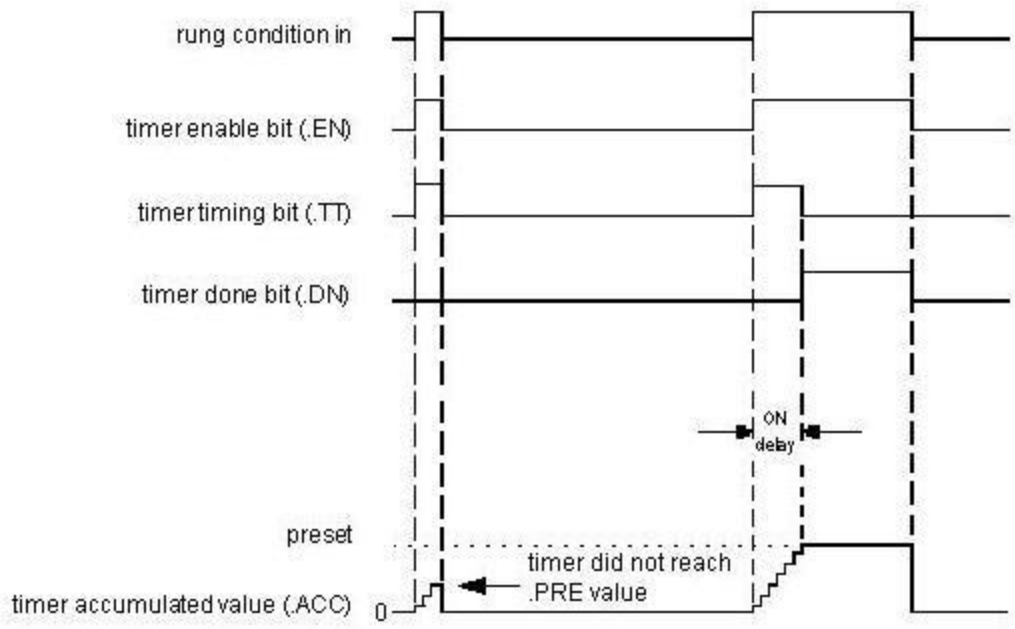
On-Delay Timer (TON)

It is used when an action is to begin a specified time after the input becomes true. Consider an example wherein a certain step in the manufacturing process is to begin 30 seconds after a signal is received from a limit switch. The 30 seconds delay is the ON-delay timer's preset value. The figure below shows a symbolic representation of the timer.

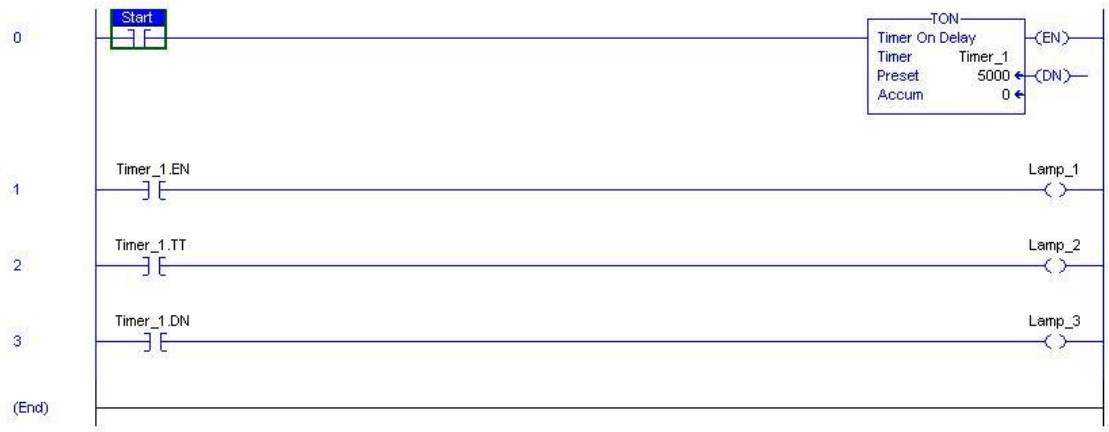
The instruction mainly includes three status bits namely EN, TT, DN. Their significance is as follows:

Enable (EN) Bit: - The enable bit indicates the TON instruction is enabled **Timer-Timing (TT) Bit:** - The timing bit indicates that a timing operation is in process. **Done (DN) Bit:** - The done bit changes state whenever the accumulated value reaches the preset value. **Accumulator (ACC) Bit:** - The accumulated value specifies the number of milliseconds that have elapsed since the TON instruction was enabled. **Preset (PRE) Bit:** - The preset value specifies the value (1msec units) which the accumulated value must reach before the instruction sets the .DN bit.

The figure shows the timing diagram which illustrates the functioning of all the bits in sequence.



The following example, after running, will illustrate the function of each bit.



Before toggling the Start, all the lamps namely Lamp_1, Lamp_2 and Lamp_3 are OFF.

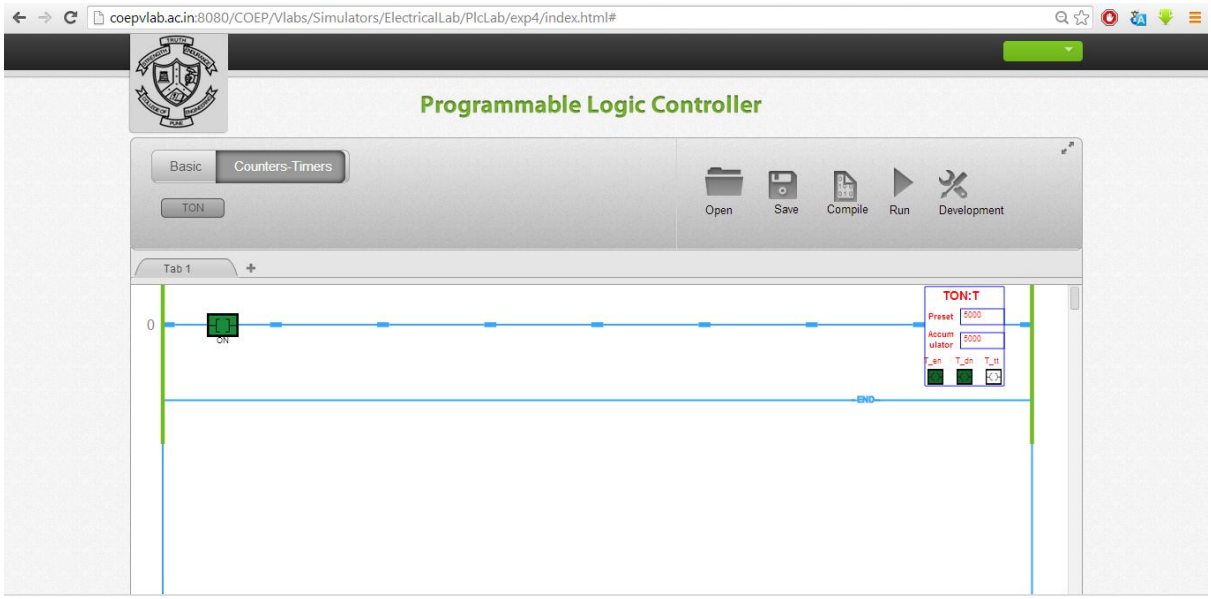
After the Start is toggled, Lamp_1 and Lamp_2 are ON. This implies the Timer_1 is enabled and its timer timing bit is activated. After the delay i.e. preset value of the timer, Lamp_1 and Lamp_3 are ON and Lamp_2 will be OFF.

The Function Block Diagram, Timing diagrams, and ladder diagram solutions are as per the available PLC(Rockwell Automation) in College of Engineering Pune.

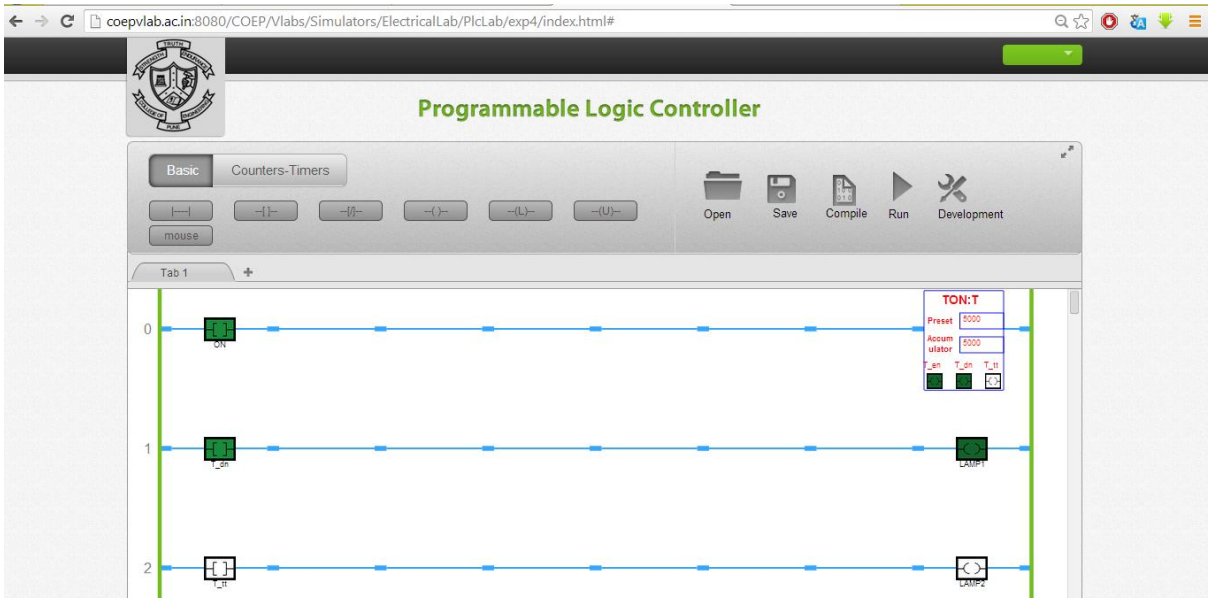
It will be better understood after the given example is developed on the simulator.

Develop an application using On-Delay Timer

- In this experiment the on delay timer will be tested for its functionalities using Simulator. Following bits of the
- timer are to be observed.
- Initialising bit "ON" in this case.
- Enable bit "T_en"
- Done bit "T_dn"
- Timer timing bit "T_tt"
- Preset value needs to be entered by the user.
- While configuring the timer thedefault time is 1 mS. Select appropriate preset value as per the need of the application. The screen shot of the configured timer will appear like this.
- To test the EN, DN, and TT bits;configure the timer by right clicking anywhere on the timer block. Submit tag and preset value.



- Add new rung to test the timer status or to energies the output. You can also test the cascading of the timer using these bits.
- Observe the tag name for timer DN bit. See following screen shot to observe the output bit status when delay is over.



- Observe the bit status in Run mode when input a

Experiment-5

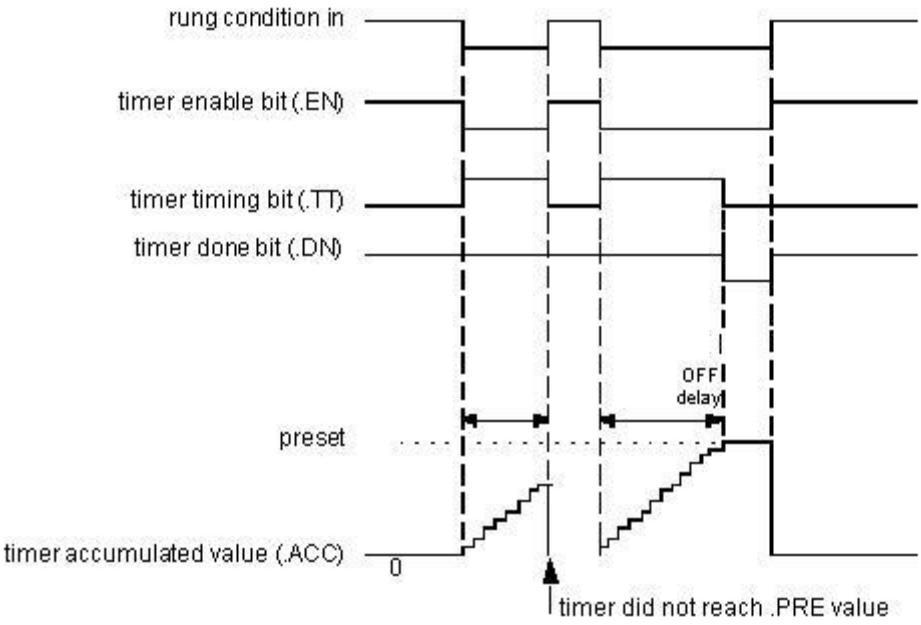
Aim: To develop an application using Off-Delay timer

Objective:

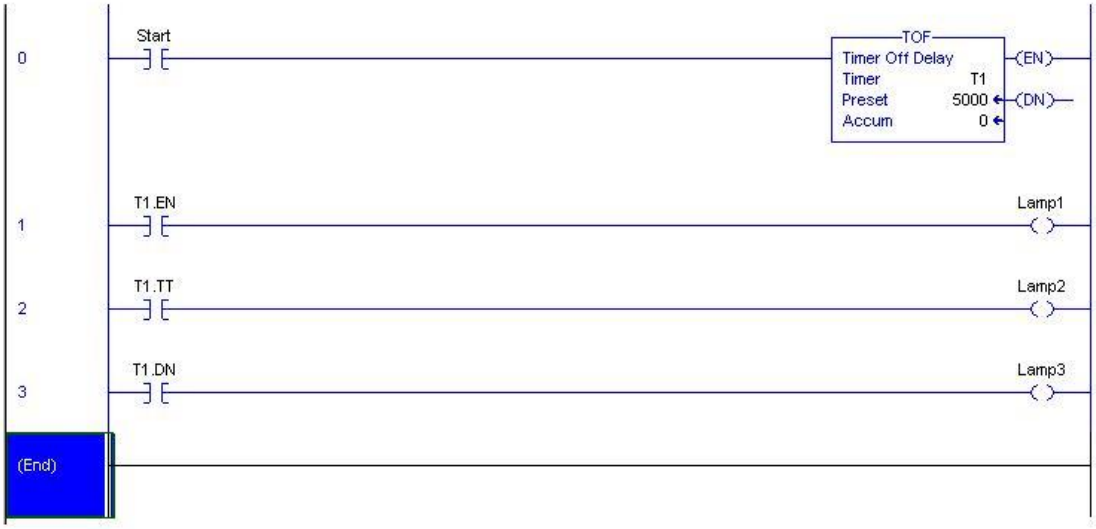
1. Study the timing diagram of OFF Delay Timer
2. Solve the assignment of Toff timer
3. **ntroduction** The timer concept and details of on delay timers with its terminology is discussed in the previous experiment. Let us learn about Off-Delay Timer (TOF). **OFF-Delay Timer:** Consider an example where the contents of a storage tank are to transfer to further process. When the low level is detected by level switch the outlet valve is to be closed. To allow entire contents to drain out, some time delay is needed as the level switch is installed slightly above the tank bottom level. This can be achieved by using off delay timer. Consider an example that, there is a Low level switch to a tank, and we have to close the drain valve of the tank after 5 second delay when low level is reached. In this case this 5 seconds delay can be given using off delay timer as we have to close the drain valve after delay. The figure below shows a symbolic representation of the off delay timer.

4.

5. The instruction mainly includes three status bits namely EN, TT, DN. Their significance is as follows:
   EN-Enable Bit: - The enable bit indicates the TOF instruction is enabled.
   TT-Timer-Timing Bit: - The timing bit indicates the timing operation is in process.
   DN- Done Bit: - The done bit changes state whenever the accumulated value reaches the preset value.
   ACC- Accumulator Bit: - The accumulated value specifies the number of milliseconds that have elapsed since the TOF instruction was enabled.
   Pre-Preset Bit: - The preset value specifies the value (1msec units) which the accumulated value must reach before the instruction clears the DN bit.
   The figure shows the timing diagram which illustrates the functioning of all the bits in sequence.
   The timing diagram illustrates the functioning of all the bits in sequence.



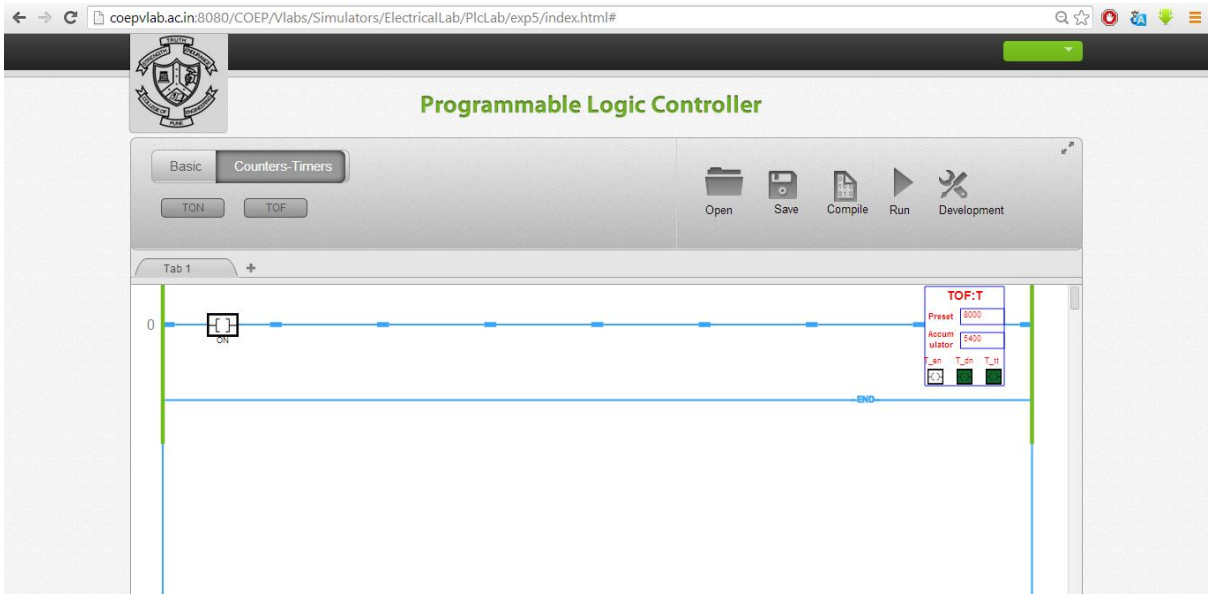
6.
7. The following example, will illustrate the function of each bit of Off Delay Timer after downloading the ladder and taking the PLC in run mode.



8.
9. For low to high transition of start bit, timer will start lamp 3 will glow, as T1.DN,bit is high. When the start bit is toggled again i.e from high to low, the delay is provided and after 5 seconds delay lamp3 will be off.
10. **The Function Block Diagram, Timing diagrams, and ladder diagram solutions are as per the available PLC(Rockwell Automation) in College of Engineering Pune.**

Develop an application using Off-Delay Timer

Implement the operation using Simulator. The configuration of off delay timer is same as 'on delay timer'. A typical difference can be observed in the operation (in Run mode) . When the q bit is energised the output DN bit goes high. The timer starts only after toggling the initialisation bit again.



Implement the operation using Simulator. To test the EN, DN, and TT bits; Configure the timer. The same tag name is to be used in the new rung to test the status or to verify the output status. You can also test the cascading of the timer using these bits.

Experiment-6

Aim: To Develop an application using UP/DOWN counter

Objective:

1. Study Counter timing diagram
2. Develop an application specific ladder program using counters

Introduction Counters are used to count number of objects or to count cycles of a typical process. Consider an example of bottle filling plant, in that counter is used to count number of bottles filled in a particular batch. In counter instruction the accumulated value will increase only when it completes the transition from open to close or vice versa. It doesn't check how long contact stay closed, it only looks for the transition. There are two basic types of counter

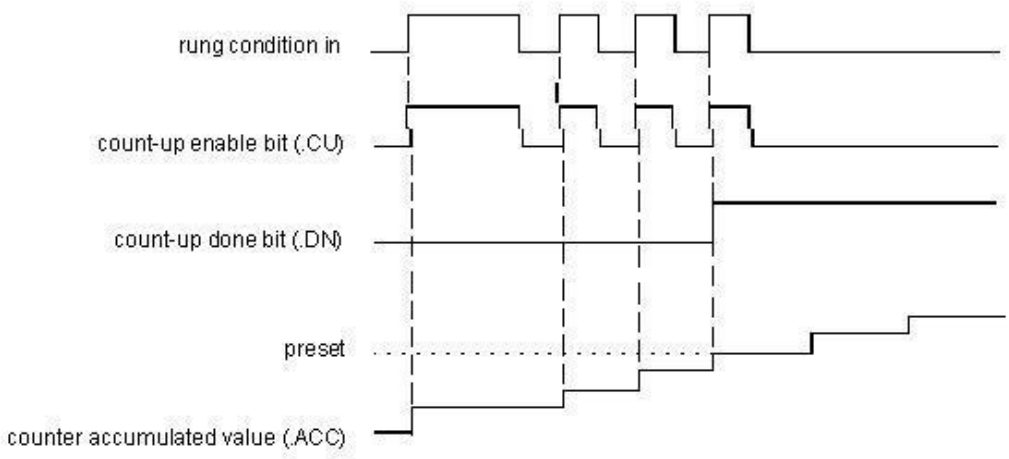
1. Up-Counter (CTU)
2. Down-Counter (CTD)

In Up counter when contact change over takes place accumulator value increments by one. While in down counter when changeover takes place accumulator value decrements by one. The instruction blocks of up counter and down counter are as shown below.

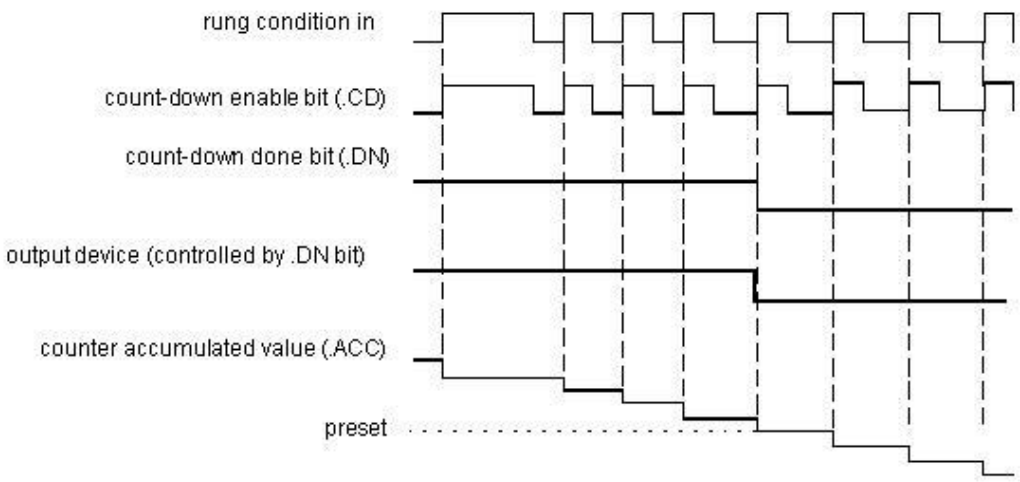
Count Up (CU) Bit: - The Count Up enable bit indicates the CTU instruction is enabled. The data type used is Boolean indicated as BOOL.
Count Down (CD) Bit: - The Count Up enable bit indicates the CTD instruction is enabled.
Done (DN) Bit: - The done bit changes state whenever the accumulated value reaches the preset value. The data type used is Boolean indicated as BOOL.
Overflow (OV) Bit: - The overflow bit indicates the counter exceeded the upper limit of 2, 147, 483, 647. The counter then roles over to -2, 147, 483, 648 and begins counting up again. The data type used is Boolean indicated as BOOL.
Underflow (UN) Bit: - The underflow bit indicates that the counter exceeded the lower limit of -2, 147, 483, 647. The counter then roles over to 2, 147, 483, 647 and begins counting down again. The data type used is Boolean indicated as BOOL.

Preset (PRE) Bit: - It specifies the value which the accumulated value must reach before the instruction sets the done bit. The data type used is Double integer indicated as DINT. **Accumulator (ACC) Bit:** - It specifies the number of transitions the instruction has counted. The timing diagram illustrates the functioning of all the bits in sequence.

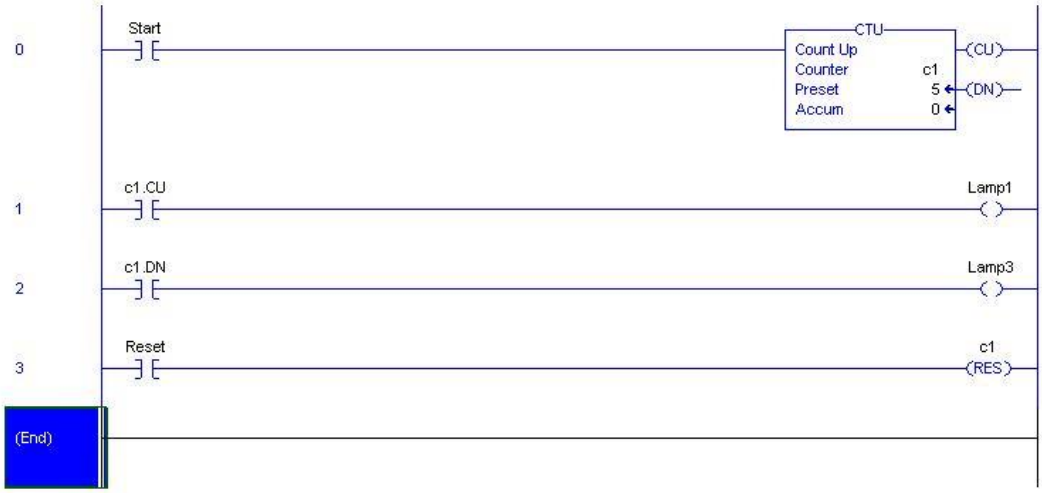
Timing Diagram for Up Counter:-



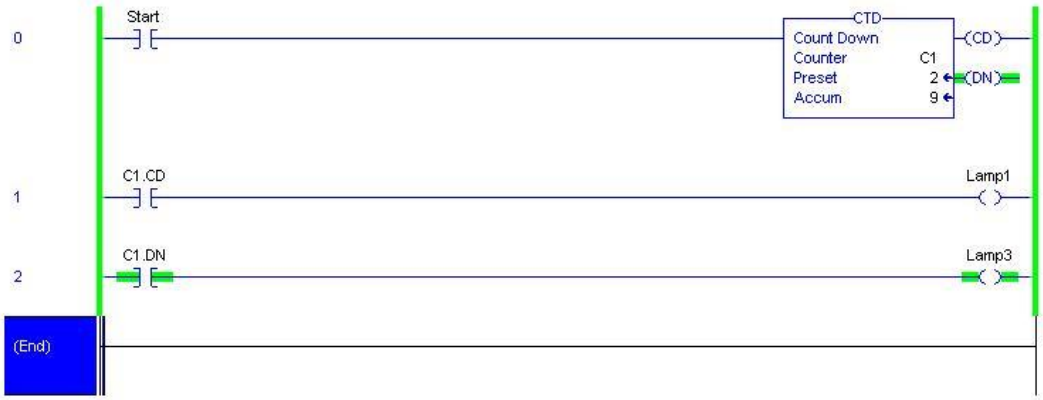
Timing Diagram for Down Counter:-



The following example, will illustrate the function of each bit of Up Counter after downloading the ladder and putting the PLC in run mode.



Here for every transition of start bit, counter accumulator value will be increased by one and for every count lamp 1 will glow on. When accumulator value becomes 5, lamp3 will glow on. To reset the counter and make accumulator value zero, reset bit is used. The following example can illustrate the function of each bit for Down Counter.

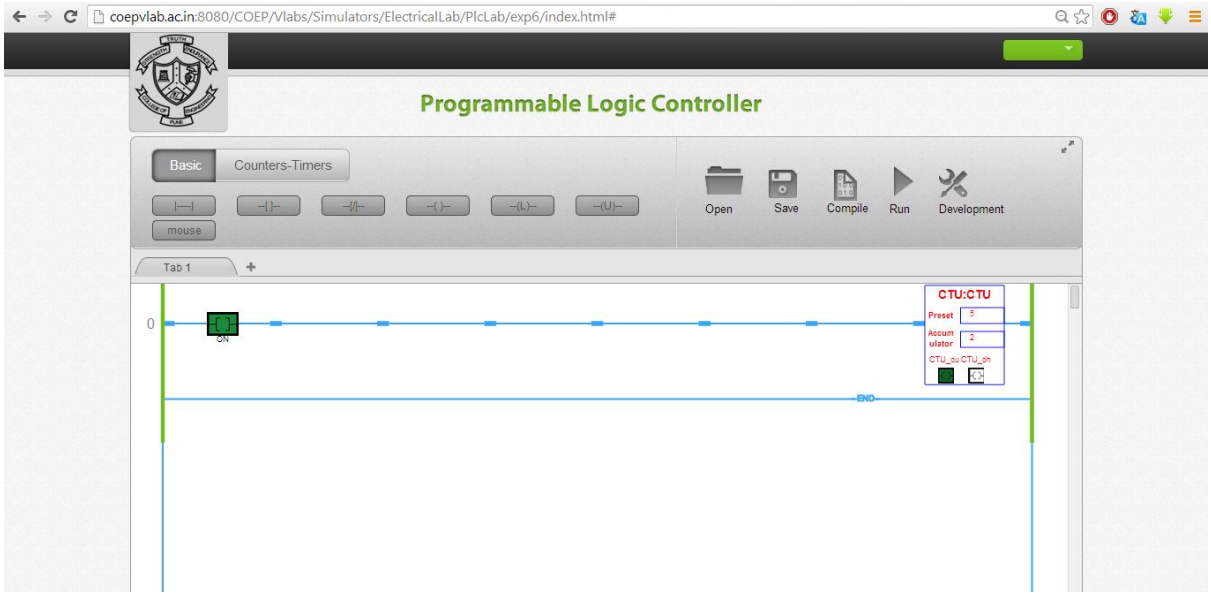


Here accumulator value of CTD is 9. For each transition of start bit this value will be decremented by one. Whenever start bit becomes on, lamp1 will glow on. Lamp3 will remain on till accumulator Value becomes greater than or equal to preset value. When it becomes less than preset then lamp3 will glow off. **The Function Block Diagram, Timing diagrams, and ladder diagram solutions are as per the available PLC(Rockwell Automation) in College of Engineering Pune.**

Procedure

Implement the counter using Simulator

The counter counts the pulses received at input. The pulses can be given by toggling the input bit "ON" in this case. The counter will keep on counting till it reaches the preset value set by the user. Once the accumulator is equal to preset the DN bit will be energised. After this instant if next pulse is detected the accumulator will increment without changing the status of DN bit. To reset the counter use "Reset" command so that the counter can be configured for new counts without reloading the Page. Please note the tag of the reset bit must be the tag of counter e.g."CTU". The screen shot will appear as shown below.



In case of down counter the entire procedure will remain same. Only the number of counts are to be entered in the accumulator tab. The preset value is zero. When the input contact closes, the accumulator will go on decrementing, will reach to zero '0' value and the status of done bit will change. To reset the DN counter use "Reset" command so that the counter can be configured for new counts without reloading the page. Please note the tag of the reset bit must be the tag of counter.

Experiment-7

Aim: To Study computational / arithmatic instructions used in PLC ladder pogramming

Objective:

1. Study Comutational Instructions available in PLC
2. Understand the use of arithmatic instructions

Theory

In case of PLC various instructions are available which can be used for computational purpose. The compute/math instructions evaluate arithmetic operations using an expression or a specific arithmetic instruction. Various instructions PLC can support are as follows.

Instruction	Description
ADD	Add two values
SUB	Subtract two values
MUL	Multiply two values
DIV	Divide two values
MOD	Determine the remainder after one value is divided by another
SQR	Calculate the square root of a value
NEG	Take the opposite sign of a value
ABS	Take the absolute value of a value

When these operations are carried out in the PLC, the type should be the same for source and destination e.g. real, integer etc. You can use mix data types, but loss of accuracy and rounding error occurs. The instruction may take more time to execute. A compute/math instruction executes once each time the instruction is scanned as long as the rung-condition-in is true. Out of the above; ADD, SUB, MUL and DIV instructions are available in the PLC simulator. The input and output parameters associated with these instructions are:

Input Parameter	Data Type	Description
EnableIn	BOOL	Enable input. If cleared, the instruction does not execute and outputs are not updated
Source A	REAL	Value to add to Source B
Source B	REAL	Value to add to Source A

Output Parameter	Data Type	Description

Output Parameter	Data Type	Description
EnableOut	BOOL	Enable output
Dest	REAL	Result of the math instruction

ADD instruction: The ADD instruction adds Source A to Source B and places the result in the Destination. When the instruction is used in Relay Ladder the output parameter conditions are defined as mentioned below.

Condition	Action
prescan	The rung-condition-out is set to false.
rung-condition-in is false	The rung-condition-out is set to false.
rung-condition-in is true	Destination = Source A + Source B. The rung-condition-out is set to true

SUB instruction: The SUB instruction subtracts Source B from Source A and places the result in the Destination. When the instruction is used in Relay Ladder the output parameter conditions are defined as mentioned below.

Condition	Action
prescan	The rung-condition-out is set to false.
rung-condition-in is false	The rung-condition-out is set to false.
rung-condition-in is true	Destination = Source A - Source B. The rung-condition-out is set to true

MUL instruction: The MUL instruction multiplies Source A with Source B and places the result in the Destination. When the instruction is used in Relay Ladder the output parameter conditions are defined as mentioned below.

Condition	Action
prescan	The rung-condition-out is set to false.

Condition	Action
rung-condition-in is false	The rung-condition-out is set to false.
rung-condition-in is true	Destination = Source A*Source B. The rung-condition-out is set to true

DIV instruction: The DIV instruction divides Source A by Source B and places the result in the Destination. When the instruction is used in Relay Ladder the output parameter conditions are defined as mentioned below.

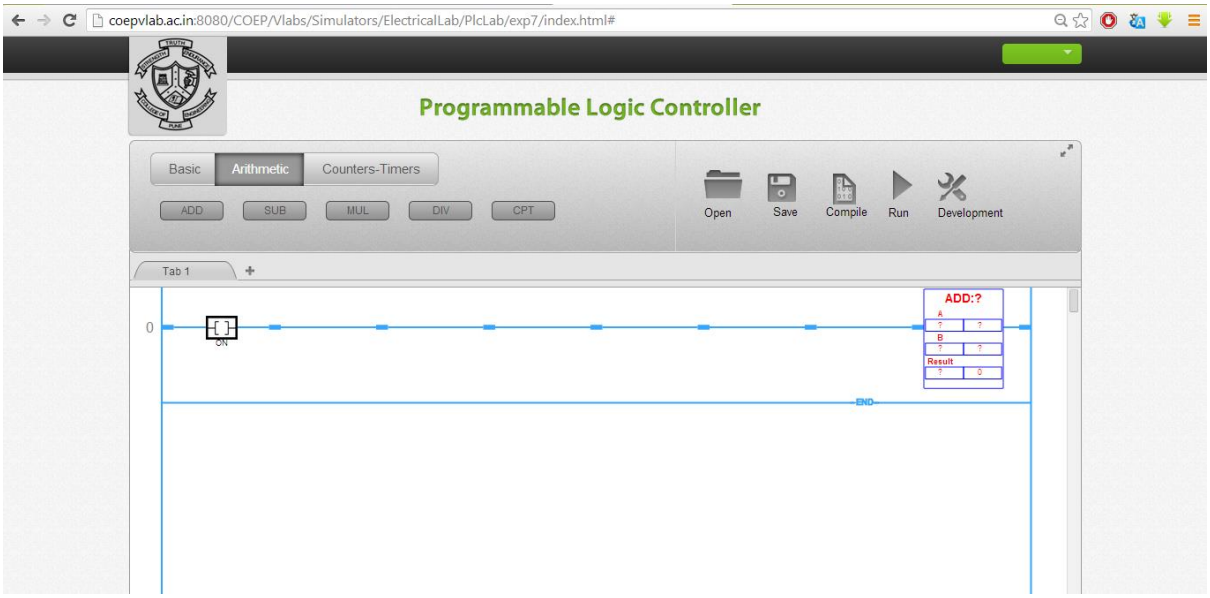
Condition	Action
prescan	The rung-condition-out is set to false.
rung-condition-in is false	The rung-condition-out is set to false.
rung-condition-in is true	Destination = Source A/ Source B. The rung-condition-out is set to true

PROCEDURE

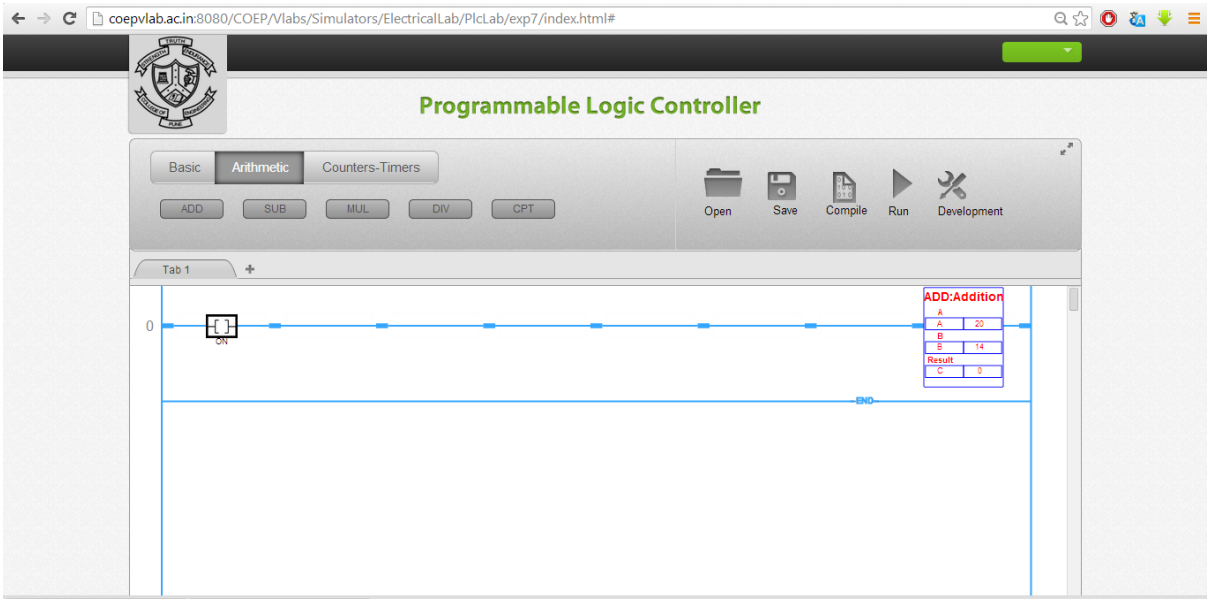
Implementation of ADD instruction using Simulator

First click on simulator tab and open the simulator by clicking on the link.

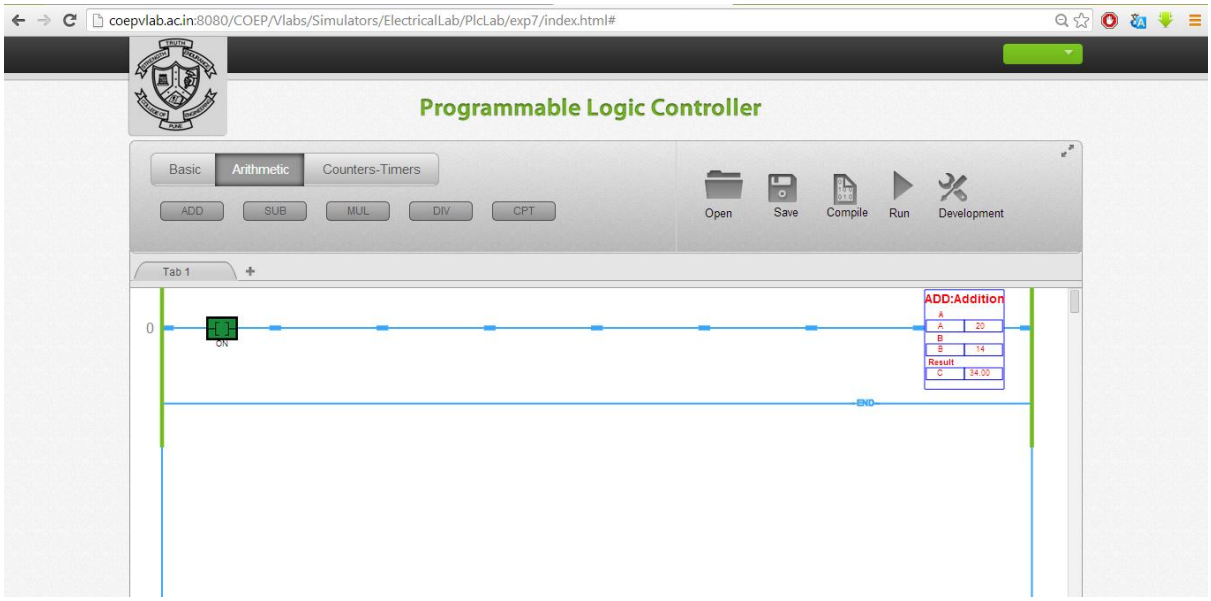
1. Add a new rung.
2. Insert NO contact. Add the ADD block provided in the simulator. The screen shot will look as follows.



3. Assign tag to NO contact and enter the addresses. Enter values for source A and B. Observe following screen shot.



4. For executing the instruction switch to run mode. Toggle the input contact and see the result at output Y as shown below.



EXPERIMENT-8

Aim: To understand working of PID function block.

Objective:

1. To understand proportional, integral and derivative control actions.
2. To study working of PID instruction using PLC simulator.
3. Observe the effect of change in Proporotional Band, Integral gain and Derivative gain values on PID performance

THEORY

Programmable logic controller were originally designed to implement discrete or logical control. But it has limited capability of handling analog input and outputs as well as analog control. PID control is available in PLC as a function. The theory behind PID control is discussed here. **In this experiment students are expected to study all features of the PID block available in PLC, in simulation mode without considering any process application.** As the name suggests this strategy is preferred for on - off type applications. This is a simplest form of control. Chattering of contacts for final control element is major problem but it can be avoided by addition of dead zone. Precise control is not possible due to addition of dead zone.
Mostly all domestic applications as Water Geyser, Electric iron, Electric Ovens are controlled using on-off control strategy.

PID: Continuous Control

There are 3 basic actions in PID.

- **P**roportional
- **I**ntegral
- **D**erivative

Proportional mode:- In this mode the controller output varies linearly with respect to the error. The equation for P mode is

```
m = Kp * e + Po
```

Where:
m is the controller output.
Kp is the gain of controller
e is the error in %
Po is the proportional Bias. (Value of m at e =0)

Due to addition of proportional bias, positive as well as negative errors are handled. Generally Po is set at 50% to handle equal positive and negative error range. Higher is the gain lower is the band to control.

Integral mode:- Integral mode is used to remove the offset produced due to P mode. Offset can either be positive and negative. Equation for Integral mode is

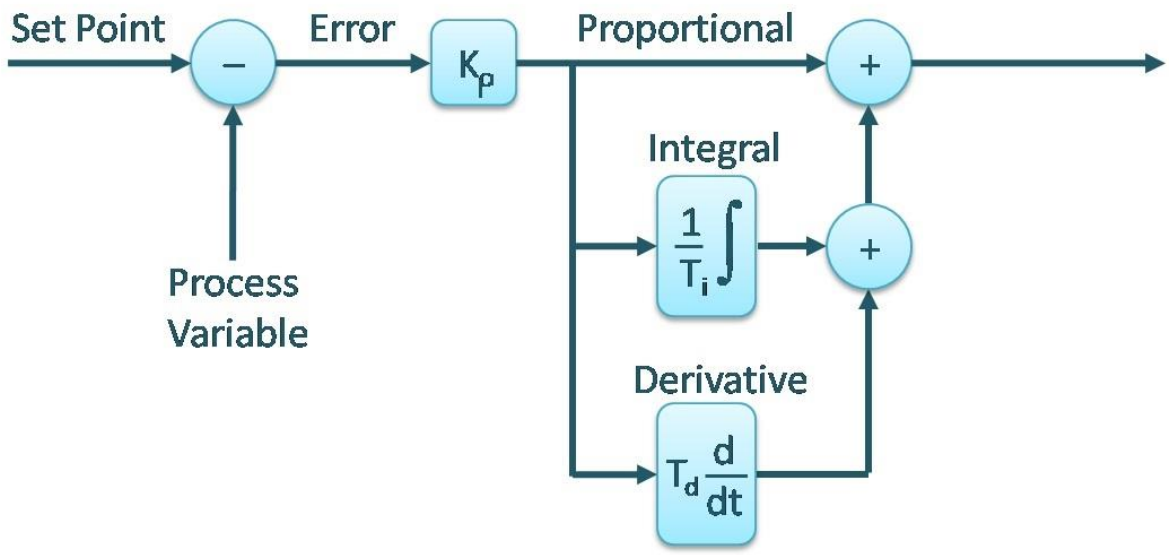
$$m = 1 / T_i * \int_0^t e dt$$

Where,
Ti is the integral time setting and t is the time.
When this action is used alone due to integral effect the action becomes too slow. When combined with Proportional, the action may go into saturation which is called as Reset Windup. To overcome this, controller output is limited at lower and higher end. This is called as Anti reset windup. But due to addition of Integral action Offset is nullified.

Derivative mode:- This action is used to increase the speed of response of slow processes. It anticipates the rate of change of error and takes the control action. Equation for Derivative mode is

$$m = T_d \frac{de}{dt}$$

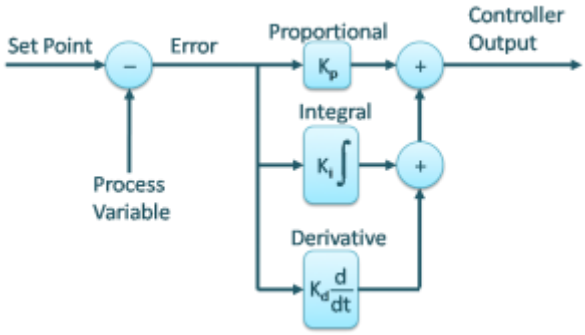
Where,
Td is the derivative time.
Set Point can be achieved quickly by using D mode. In noisy environment. this action is not used as the output goes into saturated condition. Even for constant error D action provides no correction, hence not preferred alone. It is always combined with P action. Composite controller, i.e. P, I and D adds advantages of all the three modes. The Equation for the combined controller depends on how P, I, D blocks are combined. There are two commonly used configurations, Non-interacting and Parallel. **Non-interacting PID controller algorithm:**



The equation for non-interacting controller is:

$$m = K_p\left[e + \frac{1}{T_i}\int e \cdot dt + T_d\frac{de}{dt}\right]$$

Parallel PID controller algorithm:



The equation for parallel algorithm is:

$$CO = K_p \times e + K_i\int e \cdot dt + K_d\frac{de}{dt}$$

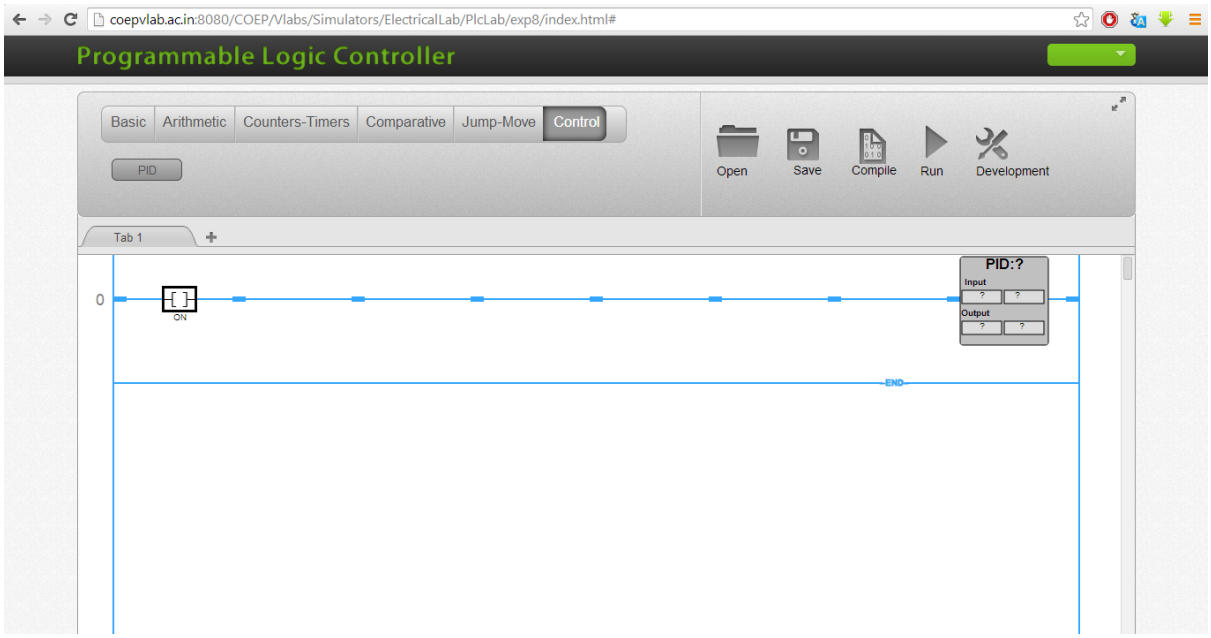
Loop tuning time can be optimized by using proper combination of PB, Ti and Td. The process parameter analog value (PV) is given as input to the block. Various settings such direct or reverse action, Proportional band setting, Set point value, Proportional band, Proportional gain, Derivative gain, Integral gain etc. are configurable parameters available in PID function.

PROCEDURE

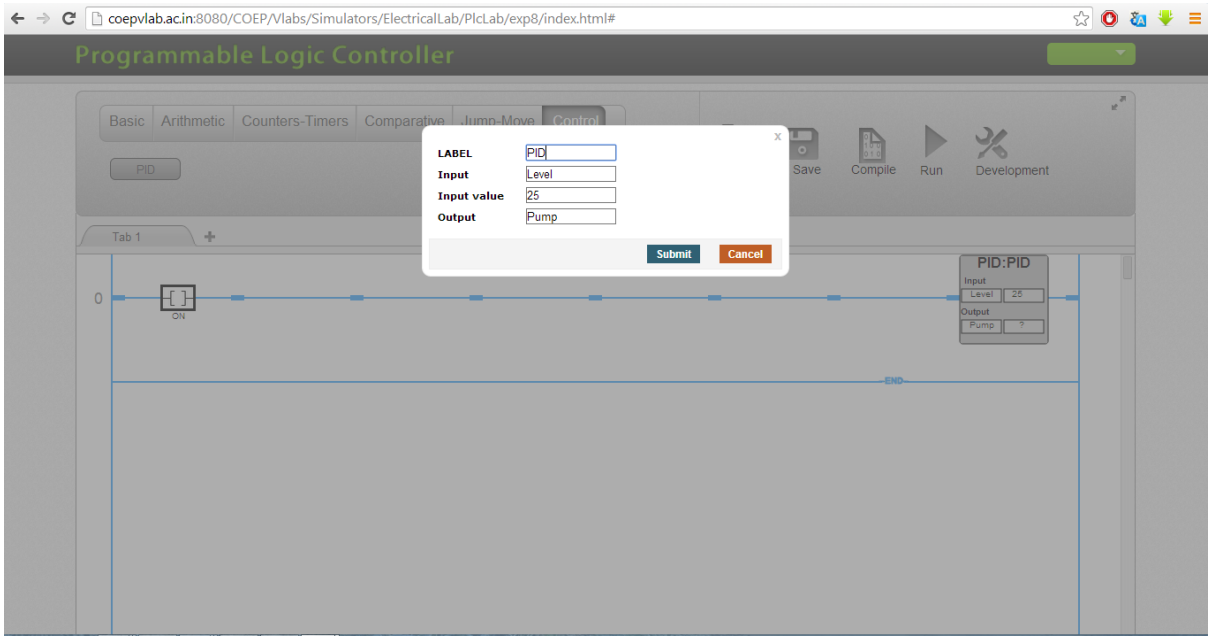
<u>Click on simulator link to understand PID opeartion.</u>

1. Add a new to rung.
2. Click on PID tab to insert PID function block in the rung.

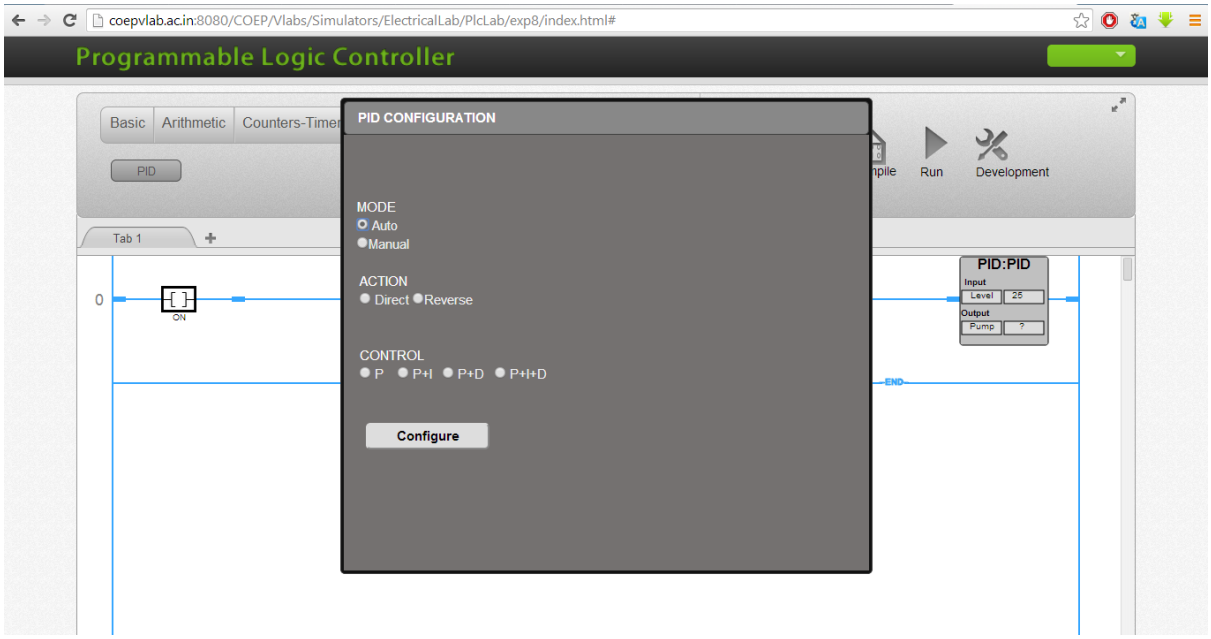
3. The screen will appear as follows.



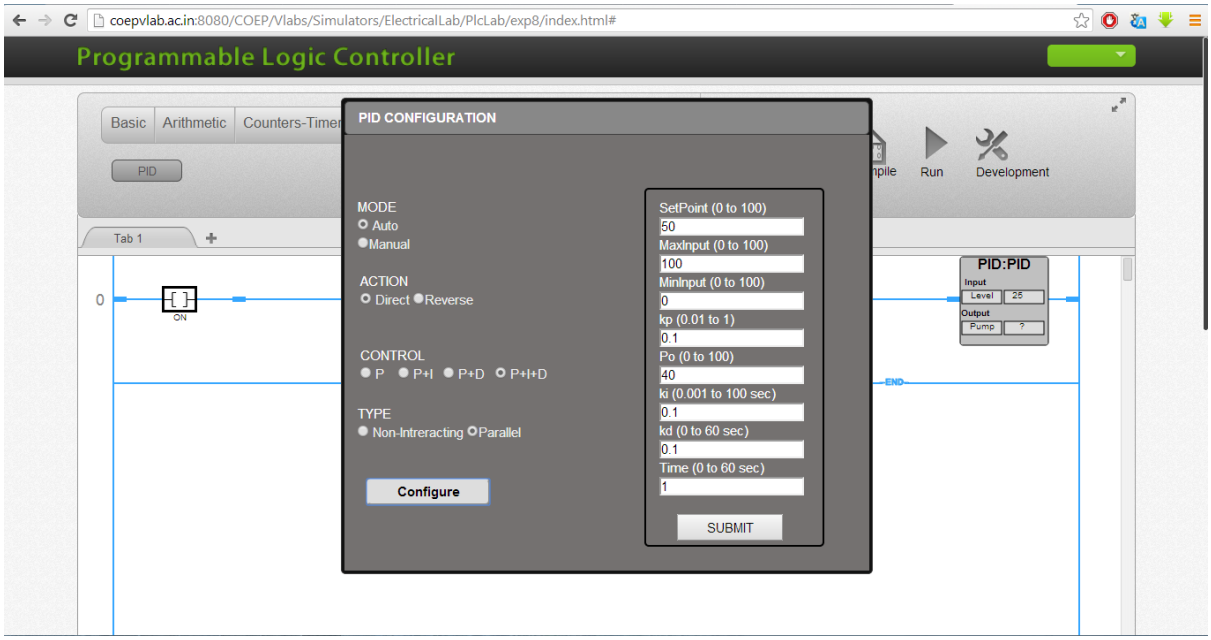
4. Now right click anywhere on PID function block to configure the bolck.



5. Once you configure the tags , You can set the action, PID mode, type of PID etc. The screen will be as follows:



6. Click on Configure tab. Now you can set the tuning parameters and submit the same. See the following screen shot.



7. Now go to run mode and observe the PID output.

Repeat steps 3 to 7 for various configurations and different tuning parameters

EXPERIMENT-9

AIM: PLC Program to Control Traffic Lights

Equipment:

1) Computer with SIMATIC software.

2) Siemens S7-1200 PLC.

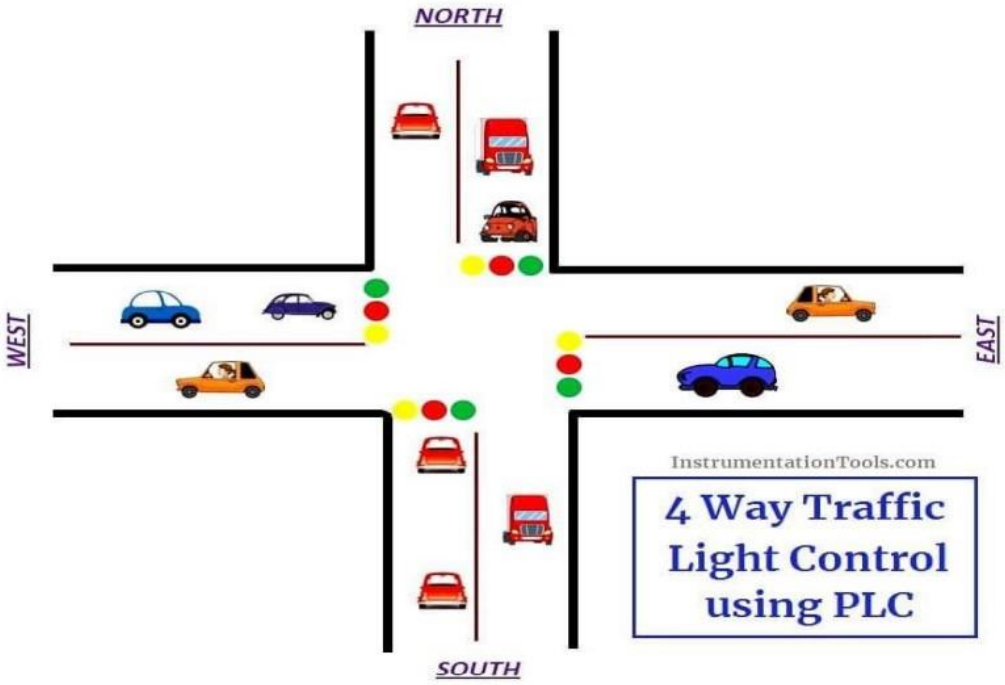
3) LEDs.

4) Switches.

5) Connecting wires

THEORY:

We most often come across four way traffic jam in our city. This PLC ladder logic gives the solution to control city traffic using programmable logic control. 4 Way Traffic Light Control.


NORTH
WEST
EAST
SOUTH
InstrumentationTools.com
4 Way Traffic Light Control using PLC

List of Inputs and Outputs

S.no	Address	Name	Input/Output
1	I:0/0	Start	Input
2	I:0/1	Stop	Input
3	B3.0	Memory	Memory
4	O:0/0	East Green	Output
5	O:0/1	North Red	Output
6	O:0/2	West Red	Output
7	O:0/3	South Yellow	Output
8	O:0/4	East Yellow	Output
9	O:0/5	North Yellow	Output
10	O:0/6	North Green	Output
11	O:0/7	East Red	Output
12	O:0/8	West Yellow	Output

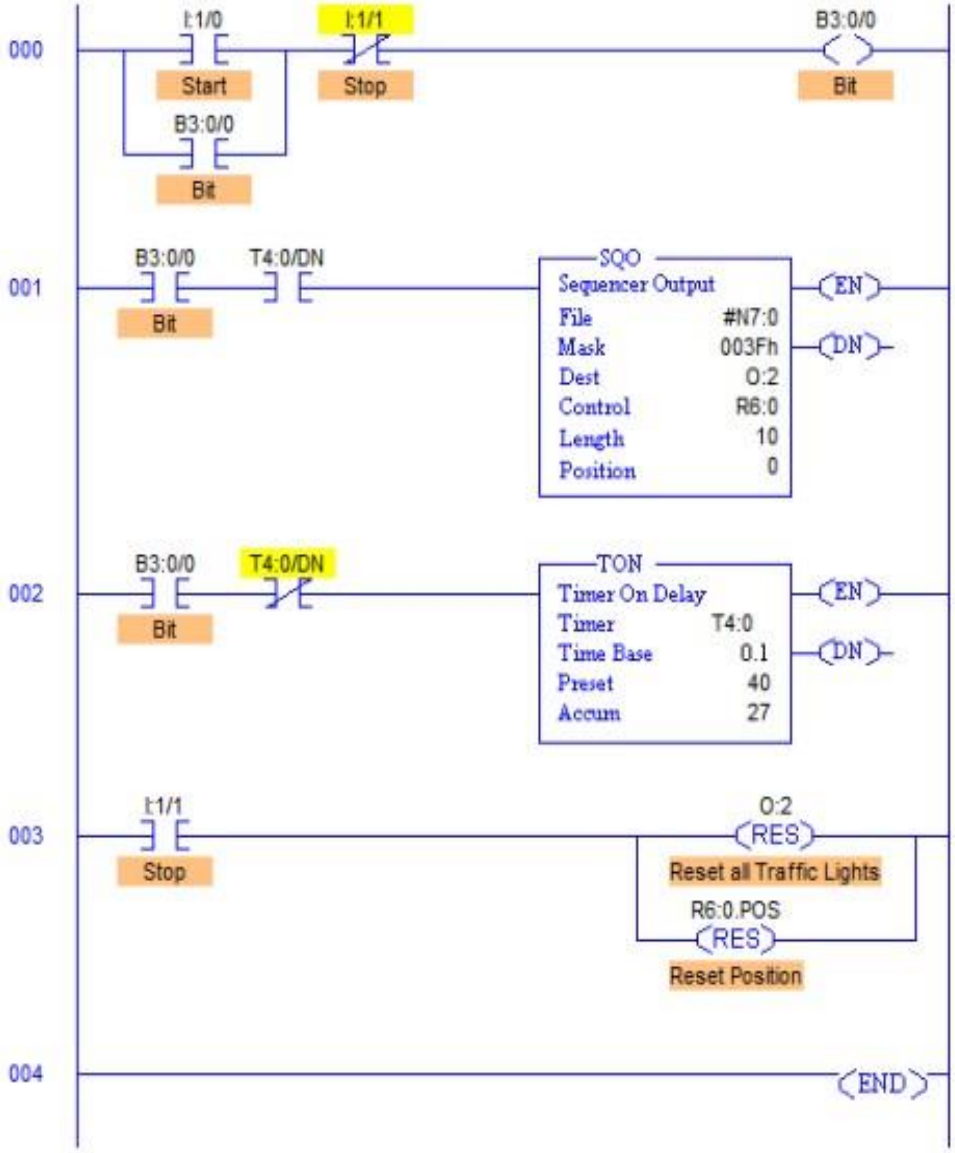
13	O:0/9	West Green	Output
14	O:0/10	South Yellow	Output
15	O:0/11	South Green	Output

Sequence of Operation

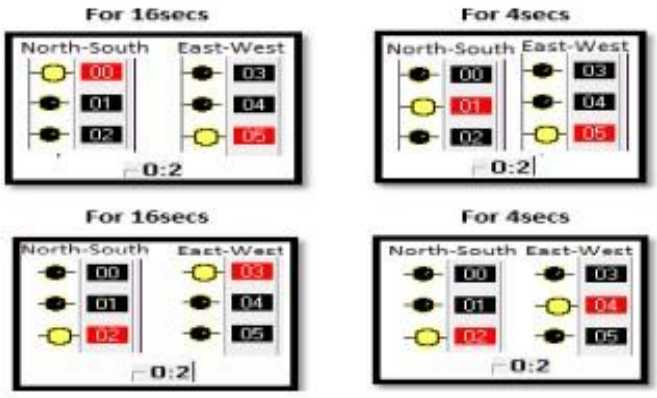
Below tabular column gives the Steps or sequence of outputs to turn ON the traffic system lamps (RED, GREEN, YELLOW)

S.NO	EAST	WEST	NORTH	SOUTH
1	G	R	R	R
2	Y	R	Y	R
3	R	R	G	R
4	R	Y	Y	R
5	R	G	R	R
6	R	Y	R	Y
7	R	R	R	G
8	Y	R	R	Y

PLC Ladder Logic



Runtime Test Cases



PROCEDURE

• RUNG000 again here is for Master Start and Stop the process.

• File; #N7:0 and File length is 10, hence output sequence is varied from N7:0 to N7:10 with each input. • Destination is set to O:2 hence with each transition, N7:0 to N7:10 are moved to O:2 with masking. • O:2/0 to O:2/5 are used as the output address to Traffic Lights and hence Mask has value 003Fh which means data flow of N7:0/0…N7:10/0 to N7:0/5…N7:10/5 is passed and the remaining N7:0/6…N7:10/6 to N7:0/15…N7:10/15 are blocked.

• Control parameters are assigned to register R6:0.

• Sequence of traffic lights to be operated are stored in the registers from N7:0 to N7:10 as following • Time base is set to 4secs, hence after every 4secs, output sequence is changed to its next register pattern outputs which is then transferred to O:2 and O:2/0 to O:2/5 are energized accordingly.

• As we can see, from N7:1 to N7:4 have the same bit pattern. So, these bits are set to 1 for 4 cycles that is 16secs. These bits are used for South-North Green light and East-West Red light.

• Similarly the entire sequence is followed.

• When Stop I:1/1 is pressed, Position is reset to 0 and all the outputs are de-energized.

Conclusion:

The above explained 4 ways traffic light control using PLC is for example only. It may vary from real time. We can use this example program to understand the working of timers and Interlocking function in PLC