Programmable Logic Controller (PLC)

Prof. Dr. M. Zahurul Haq
zahurul@me.buet.ac.bd
http://teacher.buet.ac.bd/zahurul/

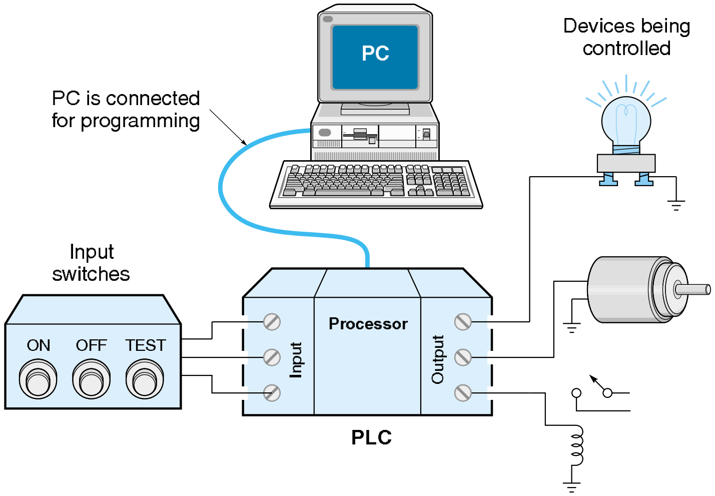
Department of Mechanical Engineering
Bangladesh University of Engineering & Technology

ME 475: Mechatronics

PLC Features

- Rugged design; suitable for harsh industrial environments against high temperature variations, dust, and vibrations.
- Industry standard I/O interfaces; capable of communicating with other PLCs, computers and intelligent devices.
- Industry standard programming languages; easily learned and understood. Programming is primarily concerned with logic, timing, counting and switching operations.
- Field programmable.
- Reduces hard wiring and wiring cost.
- Monitoring, error checking and diagnostics capability.
- Competitive in both cost and space requirements.

PLC & Related Components



e182.eps

Basic Components

PLC is essentially a microcomputer, tailored specifically for certain control tasks.

- **Hardware**: consists of the actual device technology, i.e. the PCBs, integrated modules, wires, battery, housing etc.
- **Firmware**: is the software part, known as *executive software*, that is permanently installed and supplied by the the PLC manufacturer. Programs are usually stored in ROM or EPROM.
- **Software**: is the user program. User programs are usually stored in the RAM.

Firmware

- PLC hardware does not differ significantly from computers. What makes the PLC special is the executive software. It is the internal program, provided by the manufacturer,which executes the user's program.
- The executive software determines
  - what functions are available to the user's program,
  - how the program is solved,
  - how the I/O is serviced,
  - what the PLC does during power up and down and fault conditions.

Multitasking Capability

Some PLCs are capable of executing multiple tasks with a single processor. User program assigns I/O for each task separately. Multitasking may take several forms:

- **Time driven** - it is possible to configure the processor to run each task on periodic time intervals. Hence, the time-critical job, such as the portion of that controls high-speed motions or machine fault detection, to run faster than the noncritical portions, such as servicing indicator lights.
- **Event driven or Interrupt driven** - user defines a particular event, such as an input changing state or an output turning off, that causes each tasks to run.

PLC Packaging

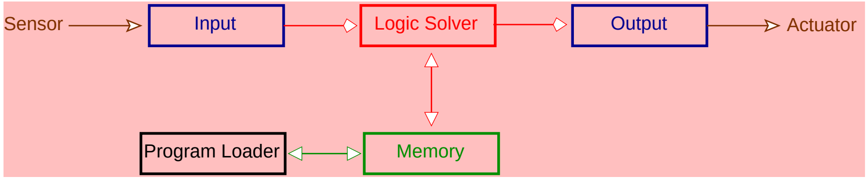
The manner in which a PLC & its I/O are packaged is critical in determining its suitability for an application.

- **Heat Removal** - appropriate means must be provided for generated heat removal to ensure low internal temperature. Commonly used methods include air venting, forced air circulation & heat sinking.
- **Mounting** - to be mounted inside NEMA rated enclosure.

Considerations in Choosing PLC

- Number and Types of **input & output** points required
- Size and type of **memory** required
- Speed and power required of **CPU** and instruction set
- **Manufacturer's support** and backup

Basic Hardware


Sensor
Input
Logic Solver
Output
Actuator
Program Loader
Memory

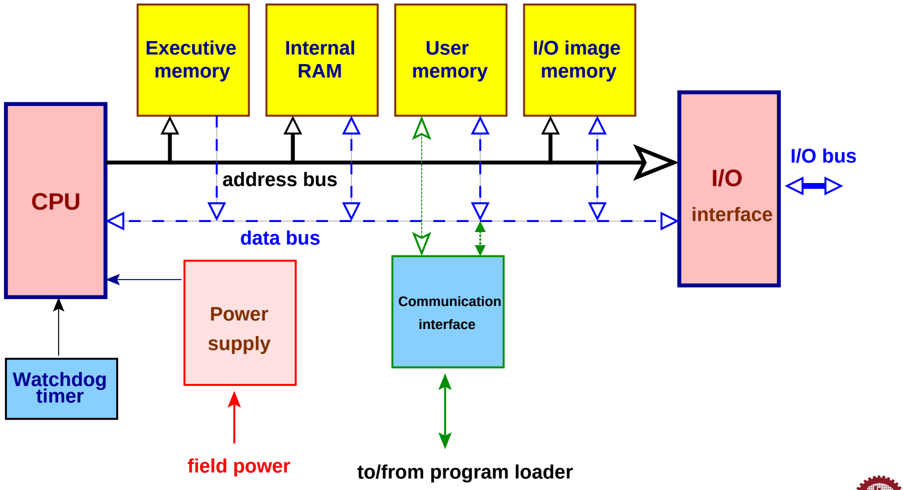
x018.eps

PLC consists of **five** major sections:

1. Power Supply
2. Memory
3. Central Processing Unit (CPU)/ Logic Solver
4. I/O Interface
5. Programming Section

- **Power Supply**- PLCs are generally powered from AC mains and power supply system converts ac voltages to required dc voltages.
- **Memory** - *Program memory* receives and holds program instructions. *Data memory* is used to temporarily hold data generated from processes or acquired through I/O devices.
- **Processor** is a micro-processor based CPU and is the part of PLC that is capable of reading and executing program instructions and data.
- **Program loader** is used to enter/change the user program into the memory and to monitor the program execution.

Detailed Block Diagram


Executive memory
Internal RAM
User memory
I/O image memory
CPU
address bus
data bus
I/O interface
I/O bus
Power supply
Communication interface
Watchdog timer
field power
to/from program loader

x019.eps

Power Supply

- Provides voltage levels required for internal operations (typically +5 V dc or ± 12 V dc).
- Provides power for I/O modules.
- Provides constant voltages.
- Packaged properly to prevent overheat.
- Separate or built into the processing unit.
- It is one of the most critical components of a PLC -
  1. It is typically non-redundant. Hence a failure of the power supply can cause the control system to fail.
  2. It usually contains high-voltage components. Hence, an isolation failure can create the potential for serious injury and fire.

Memory

The memory function of the CPU stores programs and data that the CPU needs to perform various operations. The memory is organized into several sections according to the functions they perform.

- **Executive Memory** - collection of system programs stored in ROM.
- **Scratch Pad** - is the work area used to temporarily store the binary information used by the processor. These are volatile memory as RAM-type chips are usually used. Battery backed-up CMOS RAM are also used which may last up to 10 years.
- **Processor File** - the memory block in which programmer stores and manipulates the software. The processor file is made up of *program files*, and *data files*.

Central Processing Unit (CPU)

- CPU executes a program stored in the executive memory which is set by the manufacturer.
- It organizes all control activity by receiving inputs, performing logical decisions according to the program, and control the outputs.
- CPU does not operate on the I/O directly. Rather, it works with the I/O image stored in the I/O image memory. The I/O interface is responsible for transferring the image outputs to the I/O system, reading the inputs from the I/O system, and writing them into I/O image memory.
- A 'watchdog' timer is provided to time the CPU to execute the user's program. If this time exceeds a predetermined value, watchdog timer will indicate fault and execute subsequent predefined procedure.

Input/Output (I/O) Systems

- I/O system acts as the eyes, ears and hands of PLCs.
- **Discrete I/O** - signal is discrete, such as ON/OFF, OPEN/CLOSE, energized/de-energized etc.
- **Data I/O** - complex system needs data, requires ADC/DAC.

- **Input Module** functions:
  - Reliable signal detection
  - Voltage adjustment of control voltage to logic voltage
  - Protection of sensitive electronics from external voltages
  - Screening of signals.
- **Output Modules** functions:
  - Voltage adjustment of logic voltages to control voltage
  - Protection of sensitive electronics from spurious voltages from the controller
  - Power amplification for actuation of control elements
  - Short-circuit and overload protection of output modules

I/O Systems . . .

- **Discrete I/O Inputs** - push-buttons, selector switches, joy sticks, relay contacts, pressure switches, level switches, starter contacts, temperature switches, flow switches, limit switches, photo-electric switches, and proximity switches.
- **Discrete I/O Outputs** - light, relays, solenoids, starters, alarms, valves, heating elements, and motors.
- **Data I/O Inputs** - potentiometers, temperature transducers, level transducers, pressure transducers, humidity transducers, encoders, bar code readers, wind speed transducers.
- **Data I/O Outputs** - analog meters, digital meters, stepper motors (signals), variable voltage outputs, and variable current outputs.

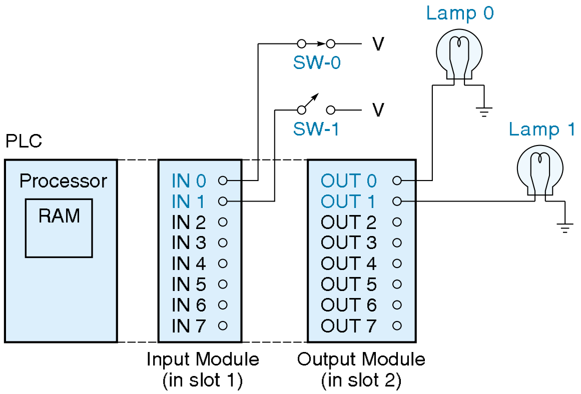
I/O Capacity

A factor that determines the size of a programmable controller is the controller’s I/O and capacity.

1. Mini-Micro – usually 32 or less I/O, but may have up to 64.
2. Small – usually 64 to 128 I/O, but may have up to 256.
3. Medium – usually 256 to 512 I/O, but may have up to 1024.
4. Large – usually 1024 to 2048 I/O, but may have many thousands more on very large units.

The I/Os may be directly connected to the PLC or may be in a remote location. I/Os in a remote location from the processor section can be hard wired back to the controller, multiplexed over a pair of wires, or sent by a fiber optic cable.

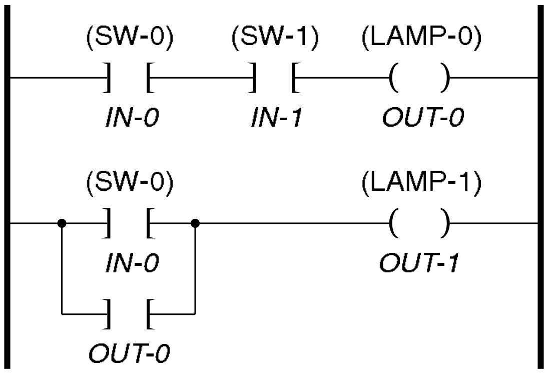
PLC Operation Review



e194.eps

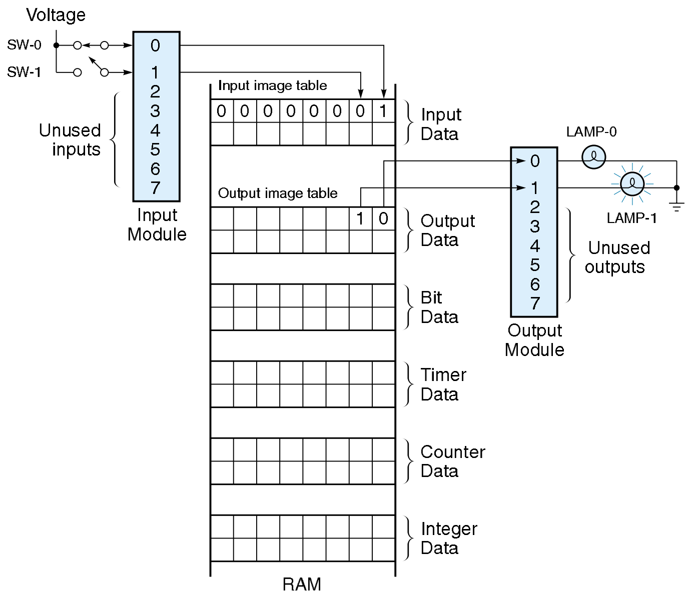
2 switches (SW-0 & SW-1) are connected via terminal IN-0 and IN-1 of input module. 2 terminals of output module (OUT-0 & OUT-1) drive 2 indicator lamps (Lamp-0 & Lamp-1)

. . .



e195.eps

The ladder diagram (LD) has two rungs. The top rung will light Lamp-0 & if both SW-0 and SW-1 are closed. The bottom rung will light Lamp-1 if either SW-0 or OUT-0 are closed.



e196.eps

...

If the self diagnostic check determine that the system is operating properly, PLC start scanning operation.

1. **Update the Input Image Table**
2. **Scan Program Instructions**
3. **Update Output Terminals**

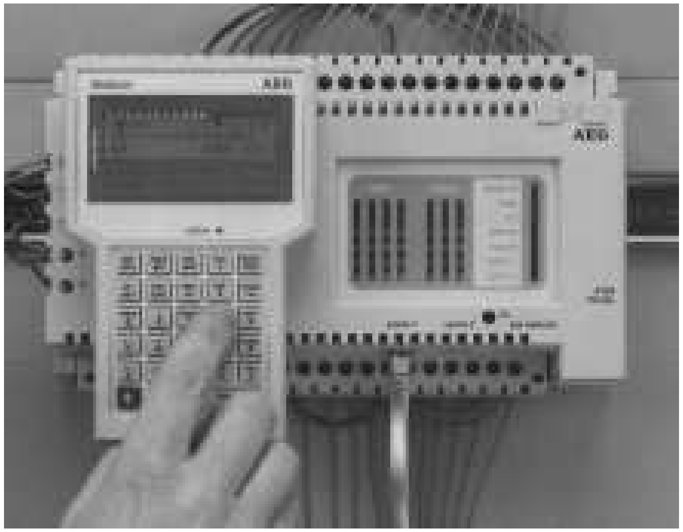
Three-step scanning process is continuous and is repeated many times each second. The time it takes to complete one scan depends on the size of the program and the microprocessor clock speed.

Programming Devices

Programming a PLC involves **3** categories:

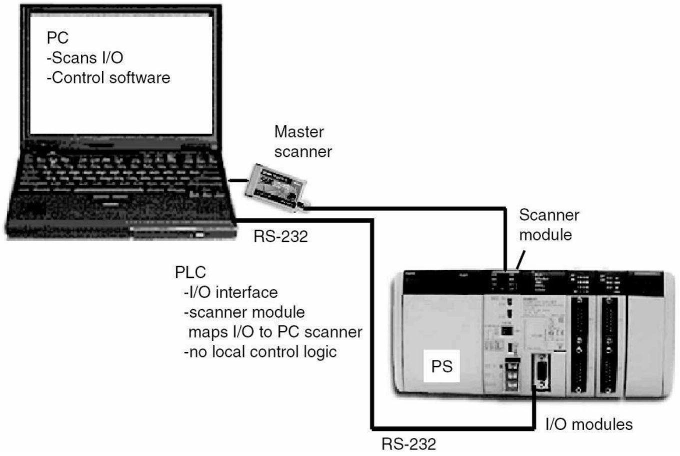
1. **Handheld Programmers** - are small inexpensive devices. These typically have membrane keys for entering data and LCD displays to show one line of a ladder program.
2. **Dedicated Terminals** - are designed for one particular brand of PLC. These provides troubleshooting operation while the PLC is running.
3. **Micro-Computers / PCs** - are widely used to program and simulate the program. Tested programs are downloaded to the PLC using serial communications.

Program Loader



e106.eps

Program Loader



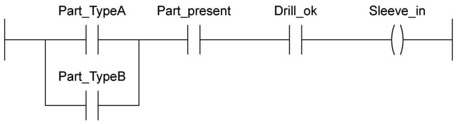
e375.eps

PLC Programming Languages

EN 61131-3 defines **five** PLC programming languages:

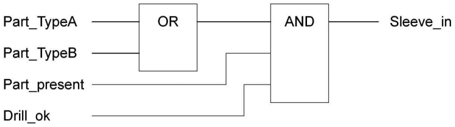
1. **Ladder Diagram (LD)**: graphic language derived from circuit diagram of directly wired relay controls.
2. **Function Block Diagram (FBD)**: functions & function blocks are represented graphically and interconnected into networks.
3. **Instruction List (IL)**: textual assembler-type language consisting of an operator and an operand.
4. **Structured Text (ST)**: high level language based on Pascal.
5. **Sequential Function Chart (SFC)**: a language resource for the structuring of sequence-oriented control programs.

Ladder Diagram (LD):



e107.eps

Function Block Diagram (FBD):



e108.eps

Instruction List (IL):

```
LD     Part-TypeA
OR     Part-TypeB
AND    Part-present
AND    Drill-ok
ST     Sleeve-in
```

Structured Text (ST):

```
Sleeve-in :  = (Part-TypeA OR Part-TypeB) AND Part-present AND Drill-ok;
```

Instruction List (IL) Mnemonics

IEC1131	Mitsubishi	OMRON	AB[1]	CH[2]	Siemens
LD	LD	LD	X10	STR/AND	A
LDN	LDI	LD NOT	XIC	STN/ANN	AN
ST	OUT	OUT	OTE	OUT	=
AND	AND	AND	AND		A
O	OR	OR	OR		O

IL Codes used differ to some extent from manufacturer to manufacturer, and IEC 1131-3 is the proposed standard to unify IL codes.

[1] Allen-Bradley
[2] Cutler-Hammer

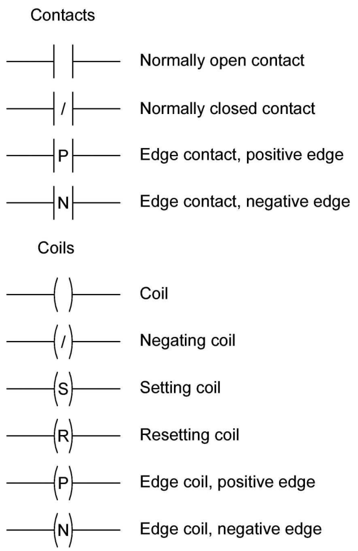
Ladder Diagram (LD)

- The use of ladder programming involves writing a program in a manner to drawing a switching circuit. The ladder diagram consists of two vertical lines representing the power rails, and circuits are connected as horizontal lines.
- Advantages of Ladder Language -
  - It is readily understood and maintained.
  - It provides graphic display of program flow.
  - Programming is fast.
  - Generates more readable programs for sequence control.

PLC Ladder Programming Conventions

1. The vertical lines of the diagram represent the power rails between which the circuits are constructed.
2. Each rung on the ladder defines one operation in the control process.
3. A LD is read from left to right and from top to bottom.
4. Each rung must start with an input or inputs and must end with at least one output.
5. Electrical devices are shown in their normal conditions, e.g. a normally closed switch is shown closed.
6. A device can appear in more than one rung of a ladder.
7. The inputs and outputs are all identified by their addresses, the notation used depending on the PLC manufacturers.

Elements of Ladder Diagram


Contacts

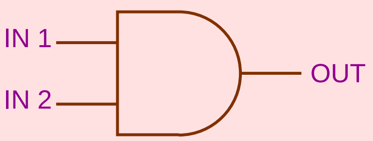
- Normally open contact
- Normally closed contact
- Edge contact, positive edge
- Edge contact, negative edge

Coils

- Coil
- Negating coil
- Setting coil
- Resetting coil
- Edge coil, positive edge
- Edge coil, negative edge

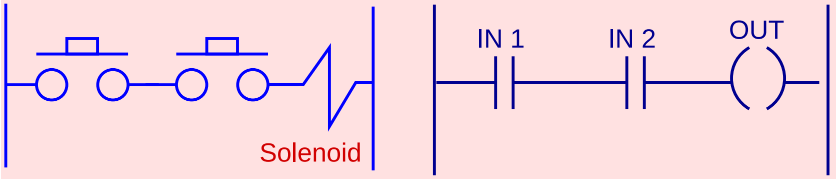
e110.eps

Logical Function: AND



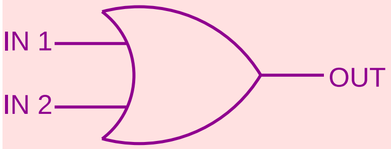
IN 1	IN 2	OUT
0	0	0
0	1	0
1	0	0
1	1	1

LD IN 1
AND IN 2
= OUT
END



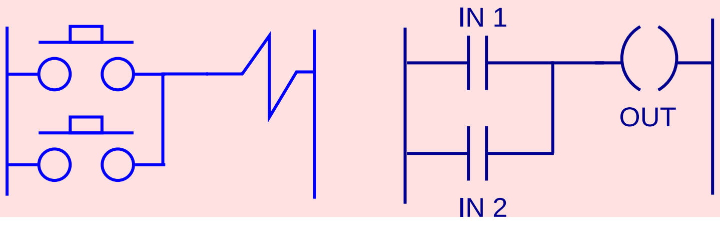
x020.eps

Logical Function: OR



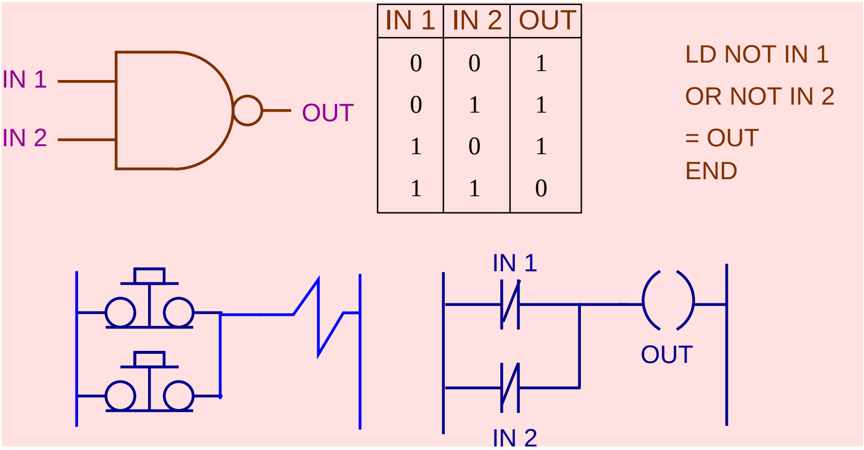
IN 1	IN 2	OUT
0	0	0
0	1	1
1	0	1
1	1	1

LD IN 1
OR IN 2
= OUT
END



x021.eps

Logical Function: NAND

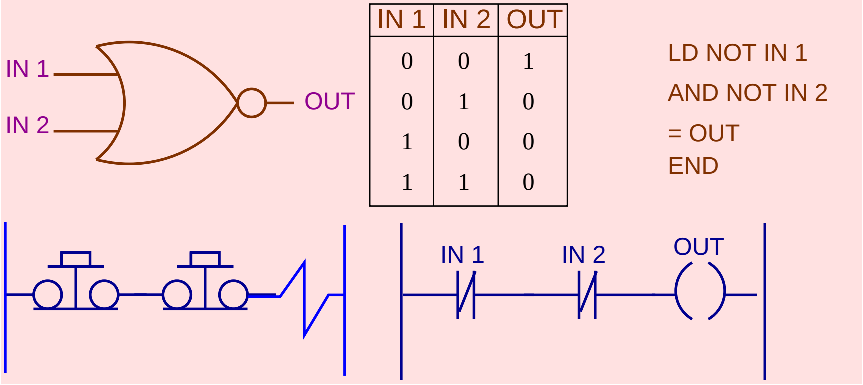


IN 1	IN 2	OUT
0	0	1
0	1	1
1	0	1
1	1	0

LD NOT IN 1
OR NOT IN 2
= OUT
END

x022.eps

Logical Function: NOR

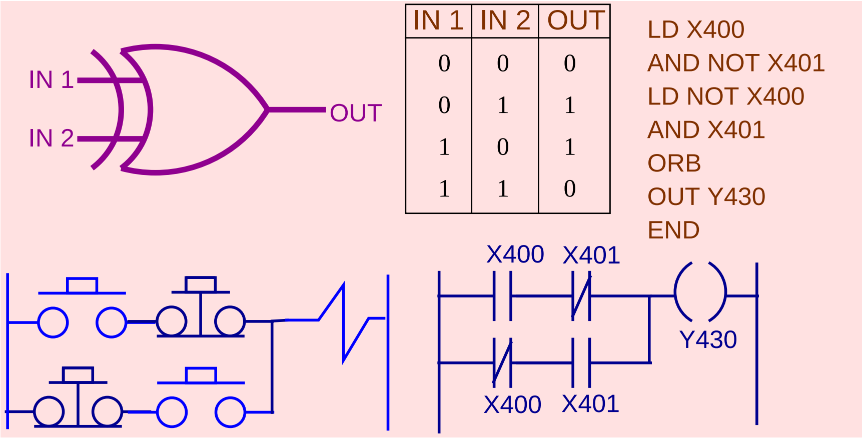


IN 1	IN 2	OUT
0	0	1
0	1	0
1	0	0
1	1	0

LD NOT IN 1
AND NOT IN 2
= OUT
END

x023.eps

Logical Function: XOR (Mitsubishi Example)



IN 1	IN 2	OUT
0	0	0
0	1	1
1	0	1
1	1	0

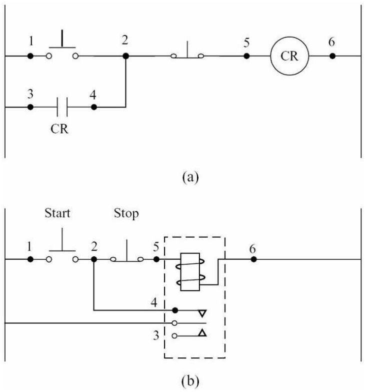
LD X400
AND NOT X401
LD NOT X400
AND X401
ORB
OUT Y430
END

x024.eps

ORB: OR branches/block together.

Latch Circuit

- After being energized, latch circuit maintains that state.
- If power fails, latch rung will be de-energized. When power is restored, machine will not automatically restarted, it can be manually restarted by pressing START switch.



(a)

(b)

e377.eps

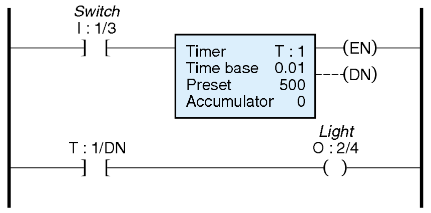
Timers/Counters

- Timers/counters instructions results in internal outputs that provides the same functions as hardware timers/counters.
- These are used to activate or deactivate a device after an expired intervals/counts.
- Both of these require an accumulator resistor to store the elapsed count/time and a register to store the preset value.
- Timers can be linked together, the term is *cascade*, to give larger delay times than is possible with just one timer.

Timers

- Timers are output instructions that are internal to the PLC. These are capable of providing timed control of devices that they activate or deactivate.
- EN 61131-3 defines **3** types of timer function blocks:
  1. **TP**: Pulse Timing
  2. **TON**: On-delay timing
  3. **TOF**: Off-delay timing
- The length of the time delay is determined by specifying a **Preset** value. Timer is enabled when the rung conditions become TRUE. Once enabled, it automatically counts up until it reaches the preset value and then goes TRUE.

Timer Instruction Parameters



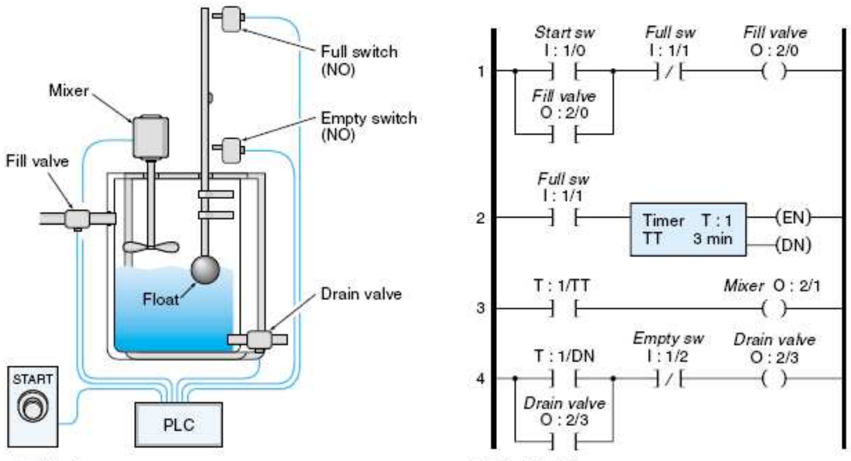
e186.eps

- **T : 1** - timer address, where 1st address hods the status bits EN, TT, & DN; 2nd address holds preset value and the 3rd address holds accumulator to hold the current value.
- **EN** - bit is TRUE as long as the timer rung is TRUE.
- **TT** - bit is TRUE as long as the timer is counting
- **DN** - bit is TRUE when the timer is done.

Timer Example

A batch process– which involves filling a container with a liquid, mixing the liquid, and draining the container – is automated with a PLC. The sequence of events is as follows:

1. *a fill valve opens and lets the liquid into the container until it is full.*
2. *liquid in the container is mixed for 3 minutes.*
3. *a drain valve opens and drains the tank.*



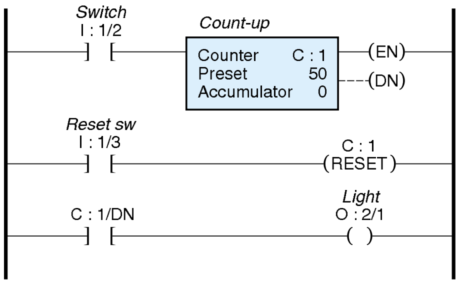
(a) Hardware

(b) Ladder diagram

e197.eps

Counters

- Counters are used to detect and count piece members and events. Counter instruction is placed in a rung and will increment (or decrement) every time the rung makes a FALSE-to-TRUE transition. The count is retained until a RESET instruction is enabled. The counter has a preset value associated with it. When the count gets up to the preset value, the output goes TRUE.
- EN 61131-3 differentiates **3** different counter modules:
  1. **CTU**: Incremental counter
  2. **CTD**: Decremental counter
  3. **CTUD**: Incremental/Decremental counter



e187.eps

- **C : 1** - counter address where 1st address hods the status bits EN, & DN; 2nd address holds preset value and the 3rd address holds accumulator to hold the current value.
- **EN** - bit is TRUE as long as the counter rung is TRUE.
- **DN** - bit is TRUE when the counting is done.
- **RESET** - when this bit goes TRUE, it resets the counter.