PROGRAMMABLE LOGIC CONTROLLERS (PLC) :

PROGRAMMING & APPLICATIONS

Learn the basic PLC and how to design PLC programming

NORAZILA BINTI MD POSDZI TAN EE CHIN

"PROGRAMMABLE LOGIC CONTROLLER (PLC) : PROGRAMMING & APPLICATIONS"

AUTHORS:
NORAZILA BINTI MD POSDZI
TAN EE CHIN

DISCLAIMER

The publisher and author have made every effort to prepare this eBook. This effort includes development, research and testing of theory to determine its effectiveness. However, all content in this eBook is intended for educational purposes only.

e ISBN 978-967-2421-35-1

Published by

Politeknik Ungku Omar
Jalan Raja Musa Mahadi
31400 Ipoh, Perak

PREFACE

Alhamdulillah

The PROGRAMMABLE LOGIC CONTROLLER (PLC) : PROGRAMMING & APPLICATIONS is specially written as a guide note for students from the Department of Electrical Engineering, Polytechnic Malaysia who take the PLC & Automation course. PLC & Automation in polytechnics is a course offered to diploma students in electrical and electronics. This eBook was created in accordance with the syllabus provided by the Curriculum Development Division of the Polytechnic Department.

Hopefully, this eBook will make it easier for users to learn PLC programming and any feedback for the future is welcome.

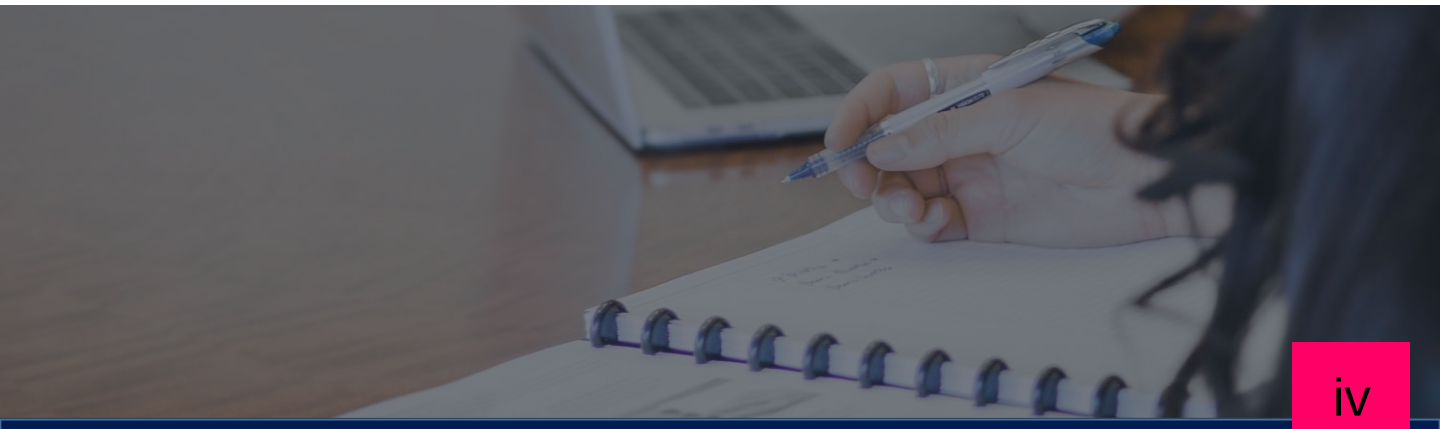
SYNOPSIS

The PROGRAMMABLE LOGIC CONTROLLER (PLC) : PROGRAMMING & APPLICATIONS is specially written for a course in PLC & Automation. The main objective in writing is to guide student understand about PLC System and how to design PLC Programming based on OMRON PLC such as CPM2A or CPM1A type.

This eBook consists of five chapters, namely:

Chapter 1 : PLC Programming
Chapter 2: Basic Logic Instruction Set
Chapter 3: Timer & Counter Instruction Set
Chapter 4: Special Instruction Set
Chapter 5: Application PLC Programming

This eBook focuses on the basic PLC system, PLC programming and PLC instruction set. Besides that, method for design simple PLC programming using a ladder diagram and instruction list (mnemonic code) are also exposed. Each sub-topic in this book is also accompanied by a self-assessment to evaluate the student learning outcomes.



ABOUT THE AUTHORS

NORAZILA BINTI MD POSDZI is a lecturer in Department of Electrical Engineering, Polytechnic Ungku Omar, Ipoh since 2016 and Polytechic Kuching Sarawak since 2009. She obtained a Bachelor's Degree in Electronics Engineering (Industrial Electronic) (Hons), UTeM, Melaka.

TAN EE CHIN is a lecturer in Department of Electrical Engineering, Polytechnic Ungku Omar, Ipoh since 1999. She obtained a Bachelor's Degree of Science in Electronic Engineering (Control), UTHM, Johor.

CONTENT

1.0 PLC PROGRAMMING

Understand PLC programming system

1. Ladder Diagram (LD)
2. Instruction List
3. Structure Text
4. Function Block Diagram (FBD)
5. Sequential Function Chart (SFC)

WHAT IS PLC PROGRAMMING?

PLC programming is an important activity that involves planning and developing programs to implement control applications based on the desired situation.

When designing a PLC program, as a programmer must know the type of PLC programming language to be used.

> PLC programming uses a programming language that is easy to use and learn by beginner.

Generally, at the international level each industry will use a different PLC programming language based on the type of PLC.

IEC 1131-3 is a global standard for control programming languages in PLCs. PLC programming languages are divided into two categories: text languages and graphic languages. The following is a list of PLC programming languages specified by this standard.

Programming Languages

Text Languages
Mnemonic Code (Instruction List)

Graphic Languages
Ladder Diagram

Graphic Languages
Sequential Function Chart (SFC)

Text Languages
Structured Text

Graphic Languages
Function Block Diagram (FBD)

LADDER DIAGRAM

Ladder diagram are a type of programming language that uses graphic methods to write control instructions.

It is the most used and best choice language for PLC programming. This language is very similar for all PLC manufacturers.

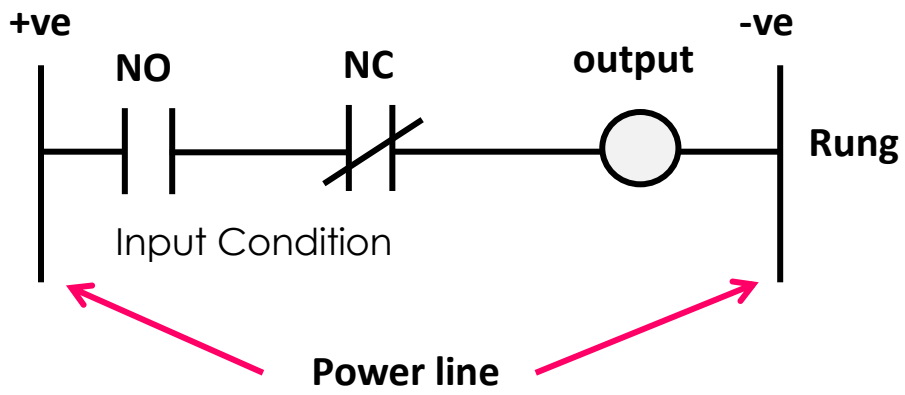


Figure 1.1 - Ladder Diagram

The ladder diagram has vertical lines on the right and left. It represents for positive (+ve) and negative (-ve) power lines, as shown in Figure 1.1.

The horizontal line that connects the bus bar is called a **Rung**, and it is where the switching elements Normally Open (NO), Normally Closed (NC) and OUTPUT are placed. Basically, a ladder diagram consists of a set of rungs in which each rung represents a single line with a specific function.

LADDER DIAGRAM

Ladder Diagram is kind of graphical programming language that evolved from the the relay control wiring circuit diagram. Figure 1.2(a) and Figure 1.2(b) shows the difference between wiring diagram and ladder diagram concept.

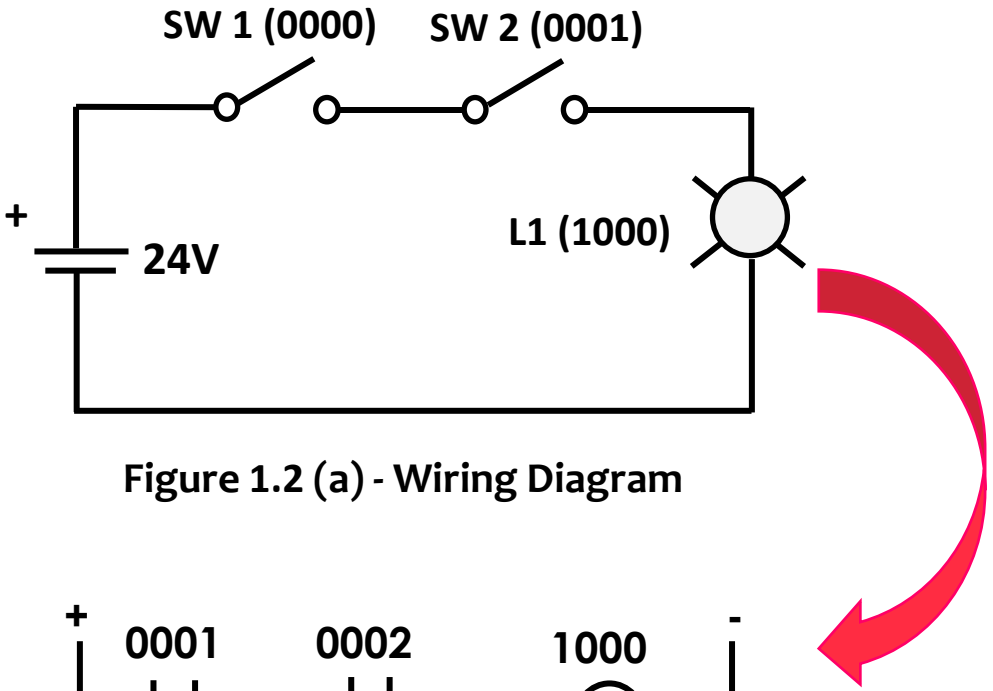


Figure 1.2 (a) - Wiring Diagram

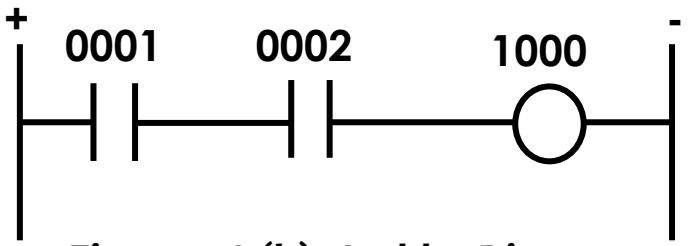
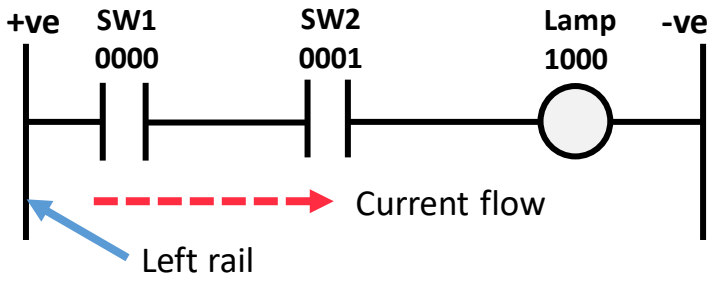


Figure 1.2 (b) - Ladder Diagram

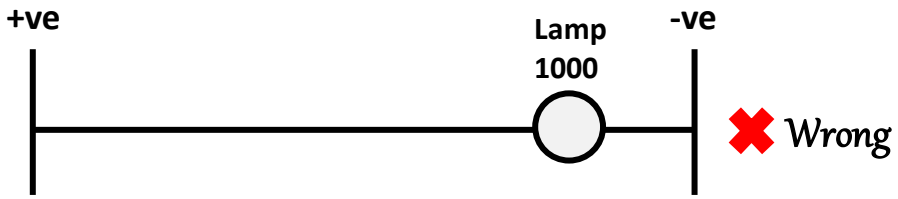
Writing this program is similar to that of drawing a switching circuit.

LADDER DIAGRAM FEATURES

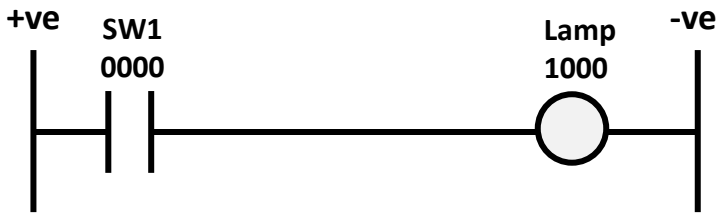
1 Power flow is from left to right (from +ve to –ve)



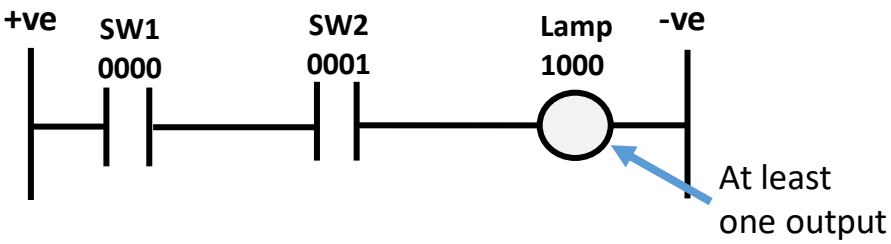
2 The output on the right can not be connected directly to the left.



There must be **at least one input** for each rung.

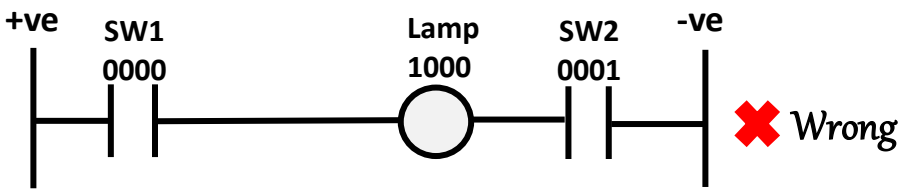


3 Each rung contains **at least one output.**

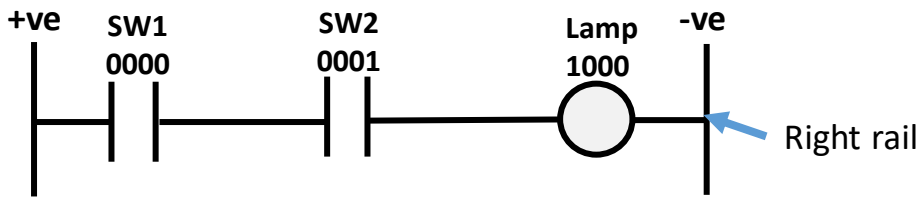


LADDER DIAGRAM FEATURES

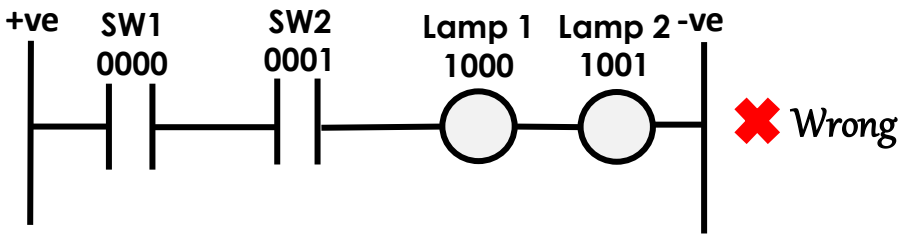
4 Contact input cannot be placed on the right side of the output.



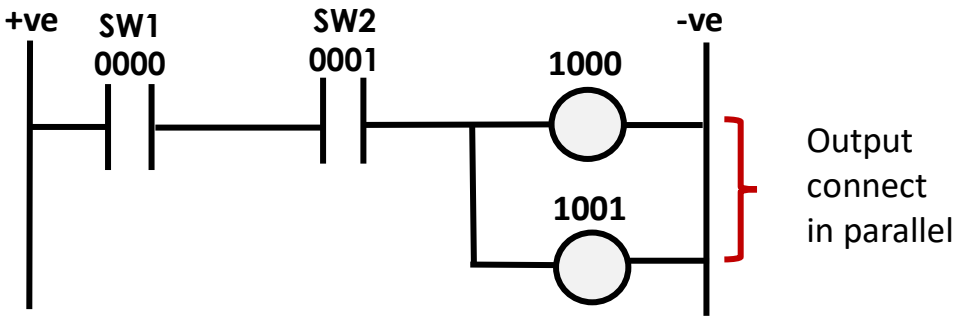
Output must connect **to a right rail.**



5 Multiple loads cannot be connected in series.

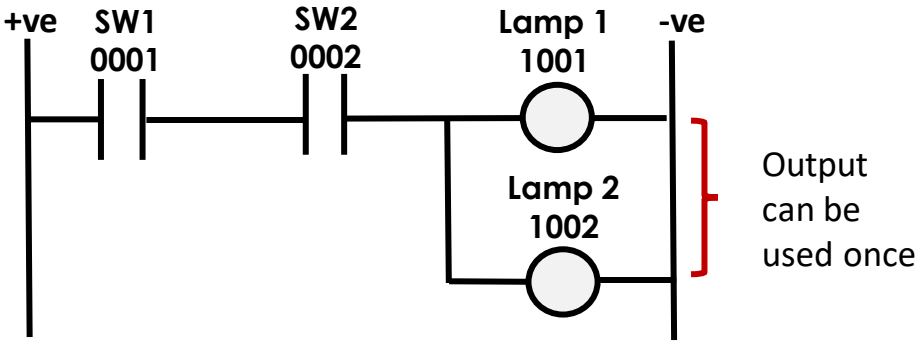


Multiple loads must be **connected in parallel.**

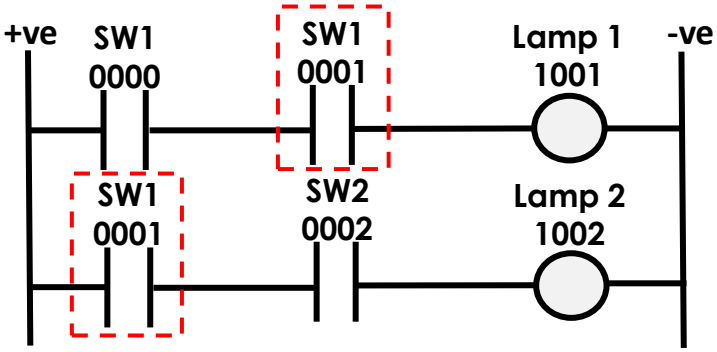


LADDER DIAGRAM FEATURES

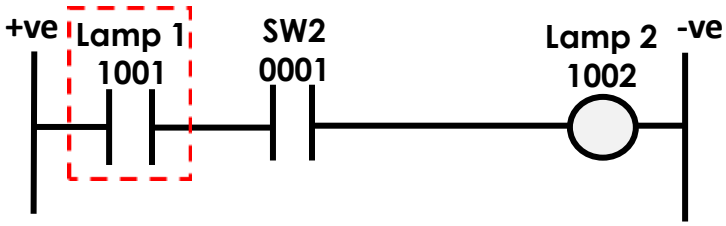
6 Each output in the programmed **can only be used once** and cannot be repeated.



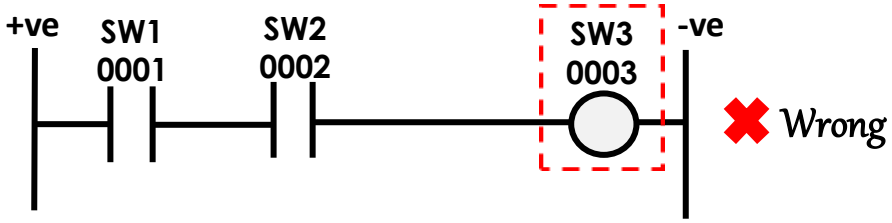
Multiple input can be used in the programmed



7 Output address **can be used** as input address



Input address **cannot be used** as an output address.



MNEMONIC CODE

Mnemonic Code (Instruction List) is a PLC programming language that uses statement lists. It is used to programmed the PLC by entering data and instructions through the Programming Console. Figure 1.3 show the instruction list operations for this programming.

Instruction List Operations

* LOAD (LD) instruction
* AND instruction
* OR instruction
* Output (OUT) instruction
* END (FUN) instruction
* NOT instruction
* AND LD instruction
* OR LD instruction

Figure 1.3

Mnemonic code is the second step after creating a ladder diagram. The ladder diagram cannot be read by the Programming Console. As a result, the Ladder Diagram should be converted to mnemonic code that contains the same information as the Ladder Diagram and can be typed directly into the Programming Console.

MNEMONIC CODE

A mnemonic code is an instruction keyword, usually an abbreviation of the actual name of the instruction.

Table 1.1 - Mnemonic Code

ADDRESS	INSTRUCTION	DATA
0000	LD	0001
0001	OR	0002
0002	AND	0003
0003	OUT	1001
0004	TIM	000
		#050
0005	FUN (01)	

Referring to the mnemonic code in Table 1.1, it consists of three column of table, namely:

Address

Consists of a four-bit number sequence, referred to as the memory address.

Instruction:

The sequence that must be followed in order to complete the task using the instruction list.

Operand/Data:

Implementation of the data input and output process. In other words, the value of the data processed by the instruction.

MNEMONIC CODE

EXAMPLE OF MNEMONIC CODE

The basic elements of a mnemonic code (instruction list) program from a ladder diagram as shown Figure 1.4 (a) and Figure 1.4 (b):

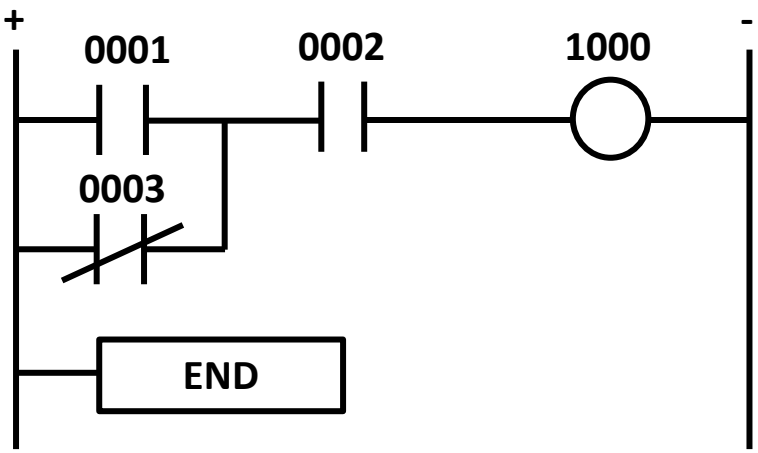


Figure 1.4 (a) - Ladder Diagram (LD)

ADDRESS	INSTRUCTION	DATA
0000	LD	0001
0001	OR NOT	0003
0002	AND	0002
0003	OUT	1000
0004	FUN (01)	

Figure 1.4 (b) - Mnemonic Code

The steps to write the mnemonic code could be found on page 25.

STRUCTURED TEXT

Structured text (ST) is a high-level text language, such as BASIC, C, or PASCAL, that has been specifically developed for industrial control applications such as process automation. It is used to implement complex procedures that are difficult to express in graphical languages.

Structured text is an extremely flexible programming language for writing control algorithms. ST programming can be written in any text editor and are easy to programmed, debug, test, and understand. Therefore, it is ideal for complex mathematical, algorithmic or decision-making tasks.

Benefits of Structured Text:

Programs run as fast and efficient as ladder diagram.

01 **Programmers can easily learn structured text programming.**

02 **Programs can be created in any text editor.**

03 **Programs run as fast and efficient as ladder diagram.**

STRUCTURED TEXT

Structured text makes use of statements to specify what should be executed.

Figure 1.5 (a) and Figure 1.5 (b) illustrates how structured text and ladder diagram programming can both be used to produce the same logical output.

EXAMPLE OF STRUCTURED TEXT

The light will turn on when one of the following two circuit condition is present:

- Switch 1 and Switch 2 switches are both closed.
- Switch 3 is closed and Switch 4 switch is open.

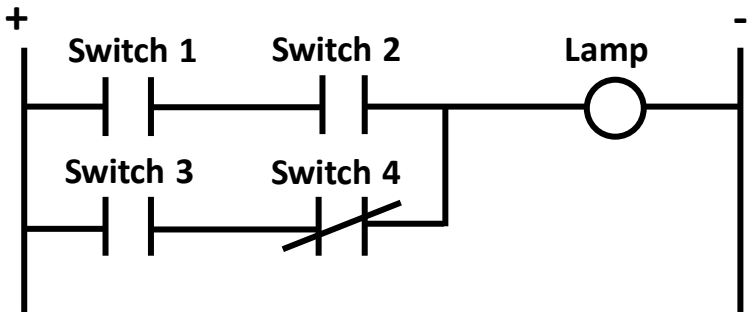


Figure 1.5 (a) - Ladder Diagram (LD)

```
IF Switch_1 AND Switch_2 THEN
        LAMP := 1;
ELSEIF Switch_3 AND NOT Switch_4 THEN
        LAMP := 1;
END_IF;
```

Figure 1.5 (b) - Structured Text (ST)

FUNCTION BLOCK DIAGRAM

Function block diagram (FBD) is used for PLC programming that use block graphs. It is a graphical language used to describe signal and data flow within a block.

The main concept of a functional block diagram is data flow. It is most useful in applications involving high information or data flow between control components, such as process control.

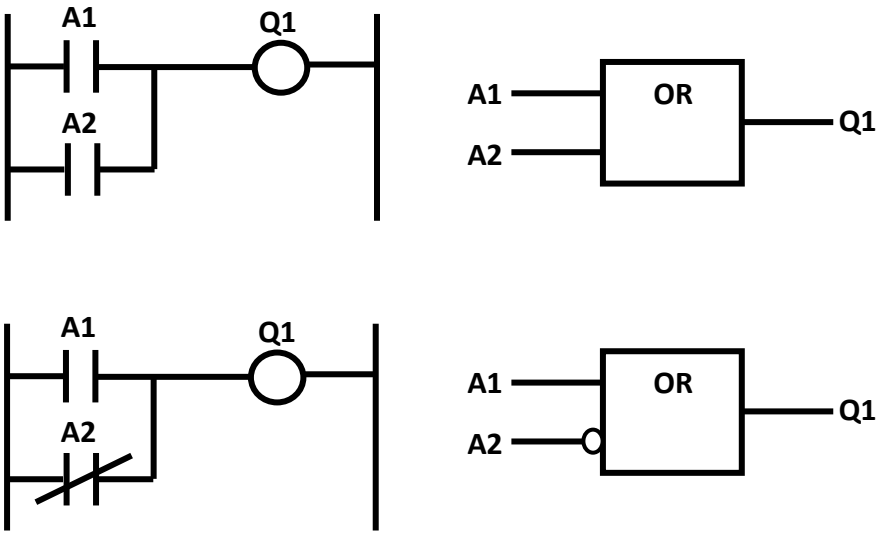


Figure 1.6 - Ladder Diagram to Function Block Diagram

Function blocks are connected to form a circuit that meets a control requirement. The block function type name, such as OR is displayed in the block as shown in Figure 1.6. Function blocks can have standard functions like logic gates, counters, or timers.

FUNCTION BLOCK DIAGRAM

The main concept is the data flow, which starts with the inputs and continues through the blocks to generate the output.

EXAMPLE OF FUNCTION BLOCK DIAGRAM

Figure 1.7 (a) and Figure 1.7 (b) shows an example between of a function block diagram and ladder diagram.

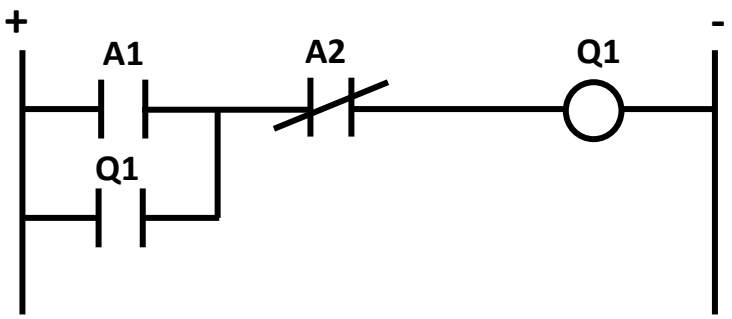


Figure 1.7 (a) - Ladder Diagram (LD)

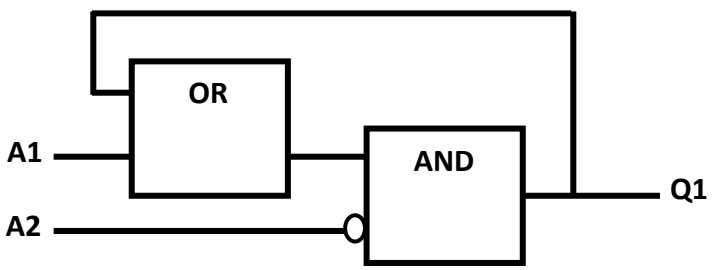


Figure 1.7 (b) - Function Block Diagram (FBD)

SEQUENTIAL FUNCTION CHART

Sequence Function Chart (SFC) is also a graphical programming language for PLC used mainly for sequential processes whereby the process can be divided into many sequential steps. These are like flowcharts, but this method is different because it does not have to follow a single path through the flowchart. Figure 1.8 shows a concept of sequential function chart (SFC).

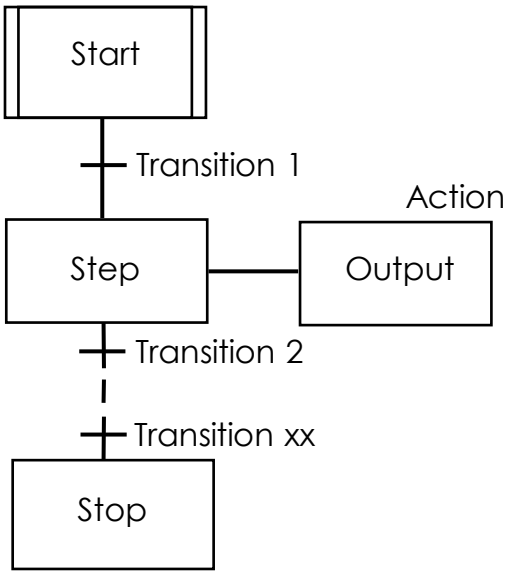


Figure 1.8 - Sequential Function Chart (SFC)

SFC programming uses a graphical interface to organize program. The three main components of an SFC are steps, actions and transitions. A step is a system function, like a mechanical process and transition is the condition needs to be fulfilled before the process can move from one step to another step.

SEQUENTIAL FUNCTION CHART

For each design using SFC, two requirements must be followed:

- There should be a transition condition between two steps.
- A step should always be between two transitions.

EXAMPLE OF SEQUENTIAL FUNCTION CHART PROGRAM

Figure 1.9 (a) and Figure 1.9 (b) shows an example between of a sequential function chart and ladder diagram program.

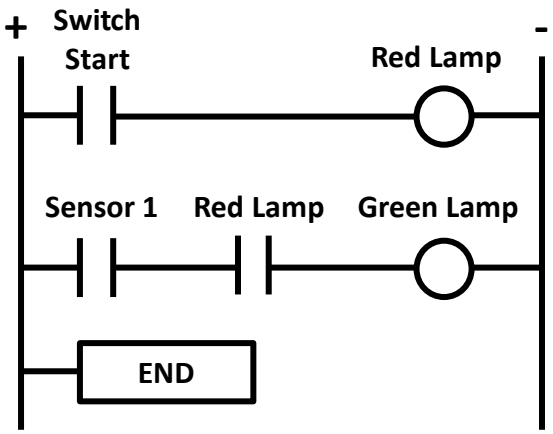


Figure 1.9 (a) - Ladder Diagram

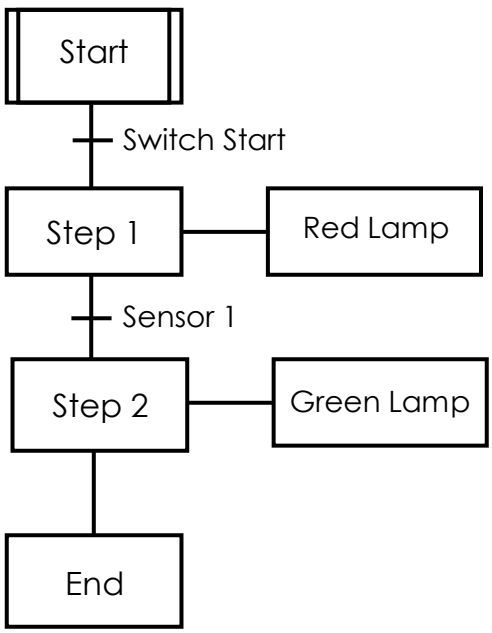
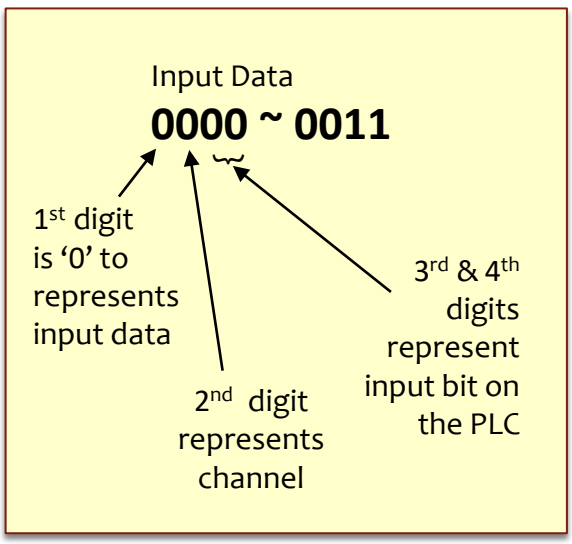
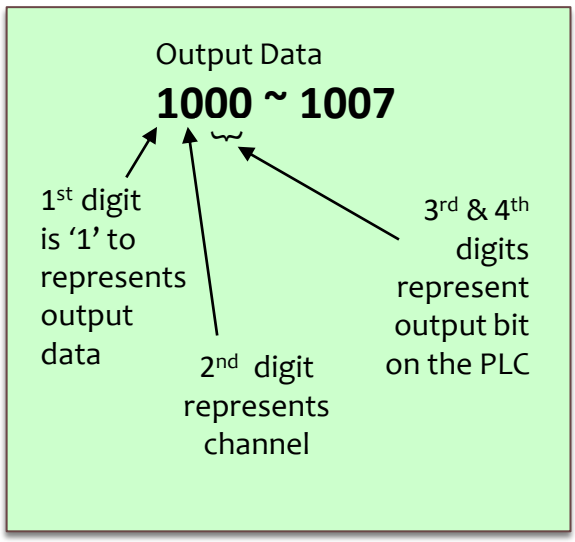


Figure 1.9 (b) - Sequential Function Chart

INPUT / OUTPUT PORT ADDRESS

In order to create the ladder diagram and mnemonic code, all external input and output devices to be connected to the PLC must be identified. In the figure below, input and output data values are described.





For example, for the Omron PLC CPM2A shown in the Figure 1.10, the input terminals are numbered from 0000 to 0011 (12 inputs), and the output terminals are numbered from 1000 to 1007 (8 output).

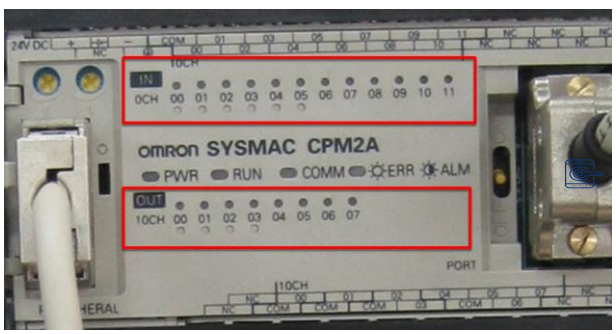


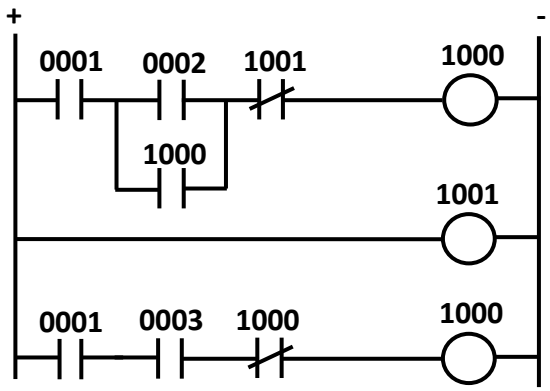
Figure 1.10 – Omron PLC CPM2A

In this E-book, All explanations are referred to OMRON PLC type CPM2A

REFLECTION QUESTION

1. List **FIVE (5)** standard PLC programming languages.

2. Based on the ladder diagram below, explain the error found in the diagram.



3. Explain the structure of the following PLC programming language:
   (a) Function Block Diagram
   (b) Sequential Function Chart

4. Convert the structured text below to PLC ladder diagram.

```
IF PB1_1 AND PB_2 THEN
        LAMP := 1;
ELSEIF PB_3 THEN
        LAMP := 1;
END_IF;
```

5. Mnemonic code consists of three column namely _____, _____and _____.

6. What is rung in PLC ladder diagram?

You can check your answer via this link or QR code

https://cutt.ly/rCF7RFR

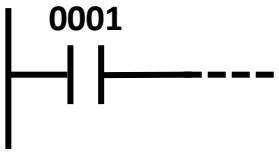
2.0 BASIC LOGIC INSTRUCTION SET

Understand basic logic instruction set

1. LOAD / LOAD NOT
2. AND / AND NOT
3. OR / OR NOT
4. AND LOAD
5. OR LOAD
6. OUT
7. END
8. No operation

LOAD (LD)

- LD is the starting instruction for the logic line of the program.
- LD is use when a rung starts with the **Normally Open (NO)** condition on the left power line.



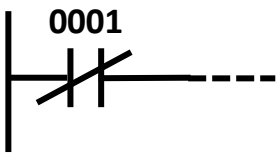
Ladder Diagram

Address	Instruction	Data
0000	LD	0001

Mnemonic Code

LOAD NOT (LD NOT)

- LD NOT is the starting instruction for the logic line of the program.
- LD NOT is use when a rung starts with the **Normally Closed (NC)** condition on the left power line.



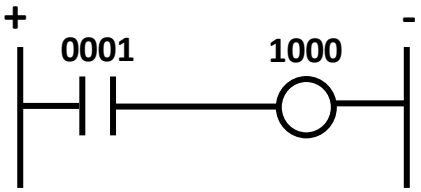
Ladder Diagram

Address	Instruction	Data
0000	LD NOT	0001

Mnemonic Code

OUT

- OUT instructions are used to control the coil output.
- OUT is connected to a right power line in the **Normally Open (NO)** condition.



Ladder Diagram

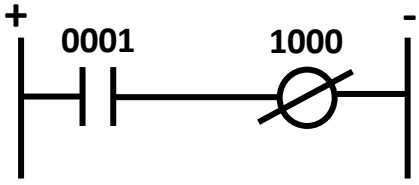
Address	Instruction	Data
0000	LD	0001
0001	OUT	1000

Mnemonic Code

Output 1000 will remain ON as long as Input 0001 is ON. When 0001 changes to OFF, 1000 also changes to OFF.

OUT NOT

- OUT NOT instructions are used to control the coil output.
- OUT NOT is connected to a right power line in the **Normally Closed (NC)** condition.



Ladder Diagram

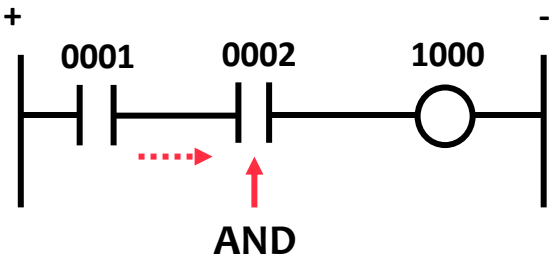
Address	Instruction	Data
0000	LD	0001
0001	OUT NOT	1000

Mnemonic Code

Output 1000 will remain ON as long as Input 0001 is OFF. When 0001 changes to ON, 1000 changes to OFF.

AND

AND instructions is used to connect two or more input with a **Normally Open (NO)** condition in **serial.**



Ladder Diagram

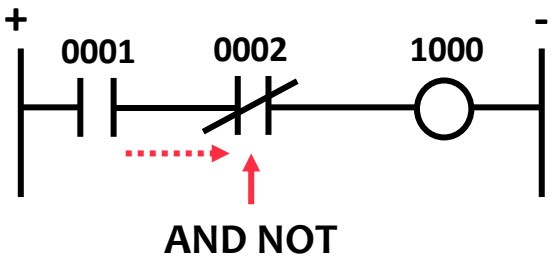
Address	Instruction	Data
0000	LD	0001
0001	AND	0002
0002	OUT	1000

Mnemonic Code

When both inputs **0001 and 0002 are turned on,** the output 1000 will also be turned on.

AND NOT

AND NOT instructions is used to connect two or more input with a **Normally Closed (NC)** condition in **serial.**



Ladder Diagram

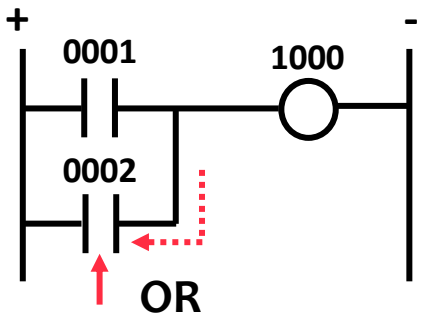
Address	Instruction	Data
0000	LD	0001
0001	AND NOT	0002
0002	OUT	1000

Mnemonic Code

When the inputs **0001 is turned ON** and **0002 is turned OFF,** the output 1000 will be turned on.

OR

OR instructions is used to connect two or more input with a **Normally Open (NO)** condition in **parallel.**



Ladder Diagram

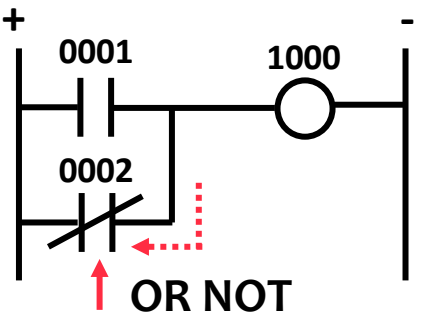
Address	Instruction	Data
0000	LD	0001
0001	OR	0002
0002	OUT	1000

Mnemonic Code

When any of the inputs **0001 or 0002 are turned on,** the output 1000 will also be turned on.

OR NOT

OR NOT instructions is used to connect two or more input with a **Normally Closed (NC)** condition in **parallel.**



Ladder Diagram

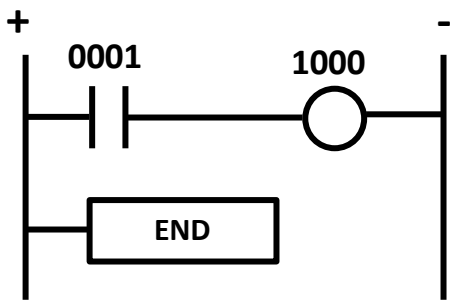
Address	Instruction	Data
0000	LD	0001
0001	OR NOT	0002
0002	OUT	1000

Mnemonic Code

When the input 0001 is turned on or 0002 is turned off, the output 1000 will be turned on. The input 0002 is NC, so the output remains on until 0002 is turned on or 0001 is turned off.

END

END instruction is used to indicate the end program. It is the last instruction of a program. The program cannot be executed without an END instruction. For OMRON PLCs, the END instruction is **FUN (01).**



Ladder Diagram

Address	Instruction	Data
0000	LD	0001
0001	OUT	1000
0002	FUN (01)	

Mnemonic Code

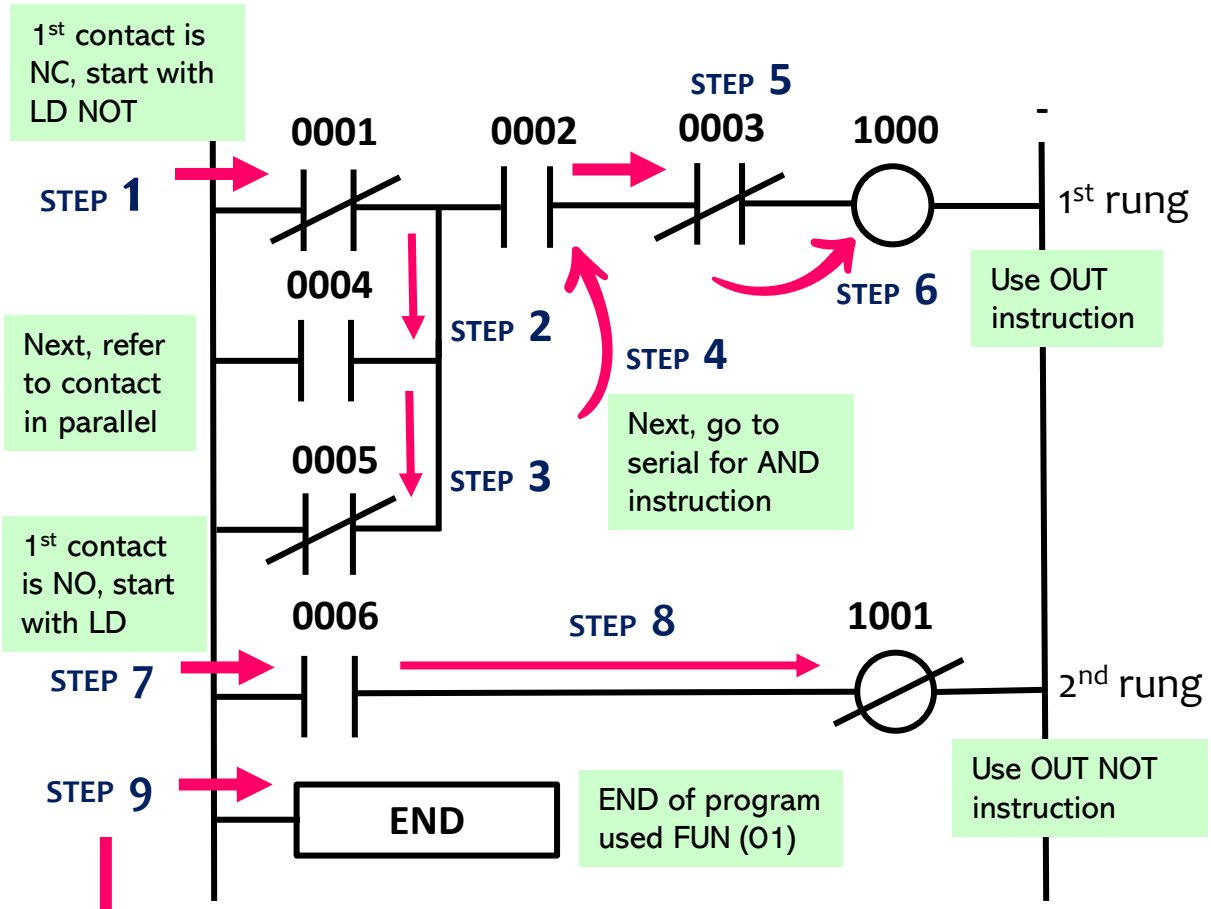
NO OPERATION (NOP)

NOP instruction will display on the PLC programming console screen when the PLC programming is deleted from the PLC's memory.

These instructions have no symbols for a ladder diagram and will not perform any operations.

HOW TO WRITE MNEMONIC CODE?

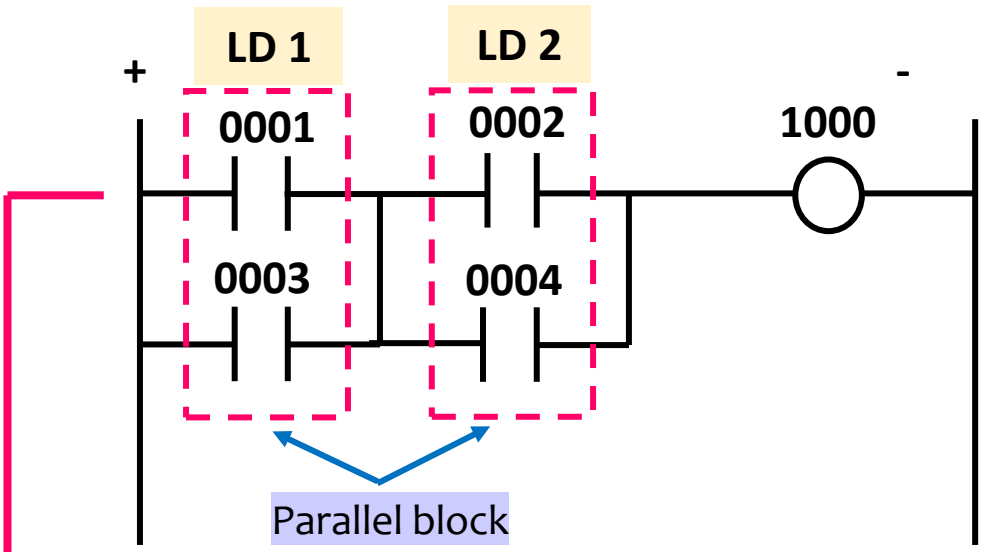
Convert from ladder diagram program to Mnemonic Code

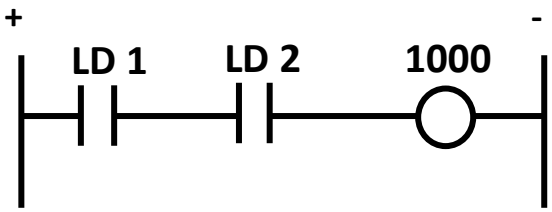


Address	Instruction	Data	STEP
0000	LD NOT	0001	1
0001	OR	0004	2
0002	OR NOT	0005	3
0003	AND	0002	4
0004	AND NOT	0003	5
0005	OUT	1000	6
0006	LD	0006	7
0007	OUT NOT	1001	8
0008	FUN (01)		9

AND LOAD (AND LD)

The AND LD instruction is used to connect two parallel instruction blocks that are connected in series.





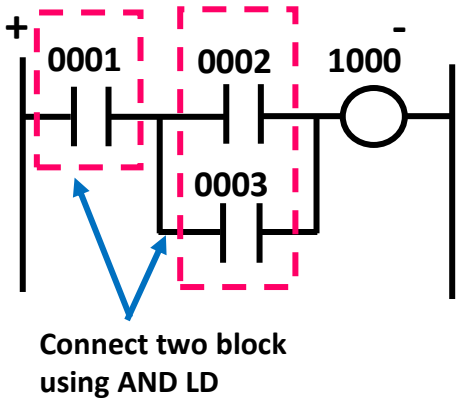
Ladder diagrams can be simplified into series connections.

Address	Instruction	Data	
0000	LD	0001	LD 1
0001	OR	0003	
0002	LD	0002	LD 2
0003	OR	0004	
0004	AND LD	-	
0005	OUT	1000	

AND LOAD (AND LD)

AND LD instruction must be used if the serial block before the parallel block.

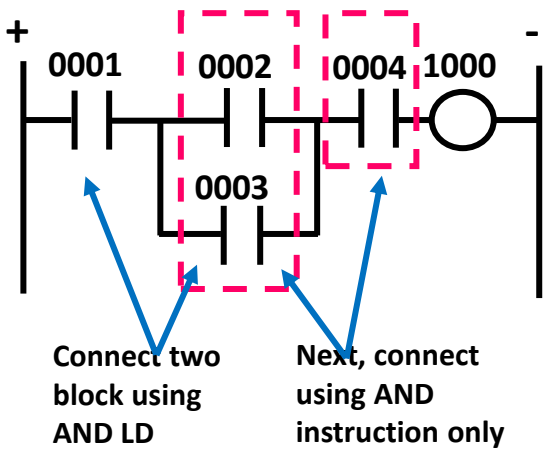
EXAMPLE 1



Address	Instruction	Data
0000	LD	0001
0001	LD	0002
0002	OR	0003
0003	AND LD	-
0004	OUT	1000

A parallel block in front of a serial block can be connected using AND connection.

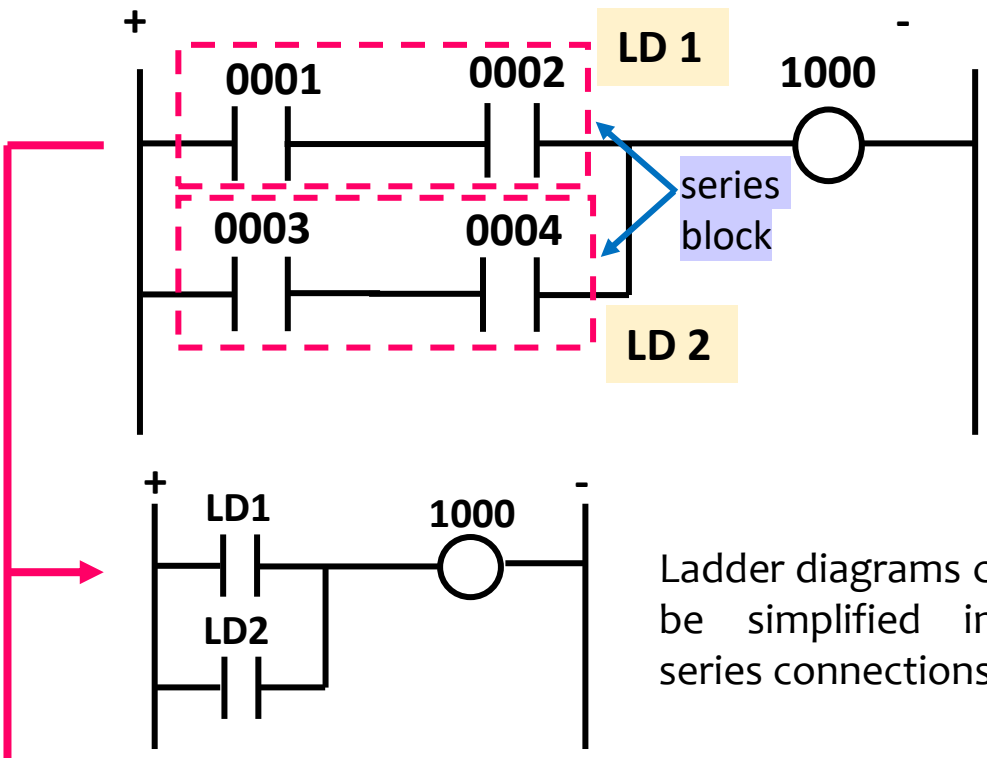
EXAMPLE 2



Address	Instruction	Data
0000	LD	0001
0001	LD	0002
0002	OR	0003
0003	AND LD	-
0004	AND	0004
0005	OUT	1000

OR LOAD (OR LD)

The OR LD instruction is used to connect two series instruction blocks that are connected in parallel.



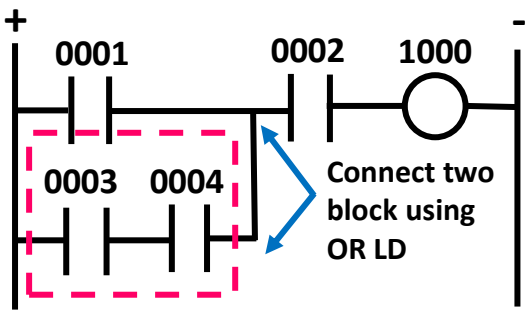
Ladder diagrams can be simplified into series connections.

Address	Instruction	Data	
0000	LD	0001	LD 1
0001	AND	0003	
0002	LD	0002	LD 2
0003	AND	0004	
0004	OR LD	-	
0005	OUT	1000	

OR LOAD (OR LD)

OR LD instruction is used if a **single contact is connected to a serial block in parallel**

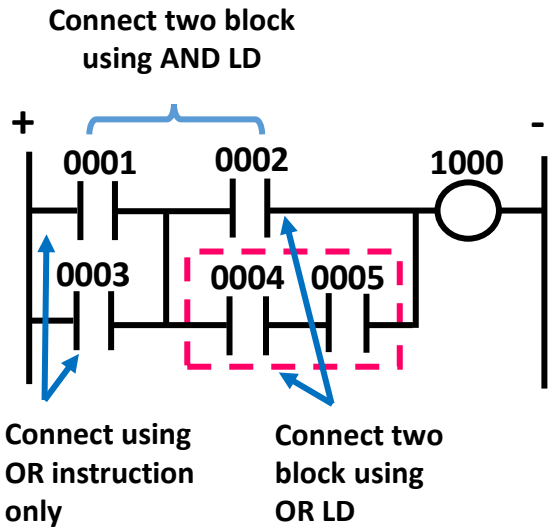
EXAMPLE



Address	Instruction	Data
0000	LD	0001
0001	LD	0003
0002	AND	0004
0003	OR LD	-
0004	AND	0002
0005	OUT	1000

COMBINATION AND LD & OR LD

The example programming using AND LD and OR LD instruction



Address	Instruction	Data
0000	LD	0001
0001	OR	0003
0002	LD	0002
0003	LD	0004
0004	AND	0005
0005	OR LD	-
0006	AND LD	-
0007	OUT	1000

TEMPORARY RELAYS (TR)

Temporary Relays (TR) are used for temporary storage of a program instructions. The program instruction found in a ladder diagram that has a special branch point as shown in Figure 2.1. The TR instruction are used for mnemonics code only.

TR instructions contain 8 bits of data and have addresses ranging from **TR0 to TR07**. TR can be used as many times as required and the number address must be the same at the same branch point.

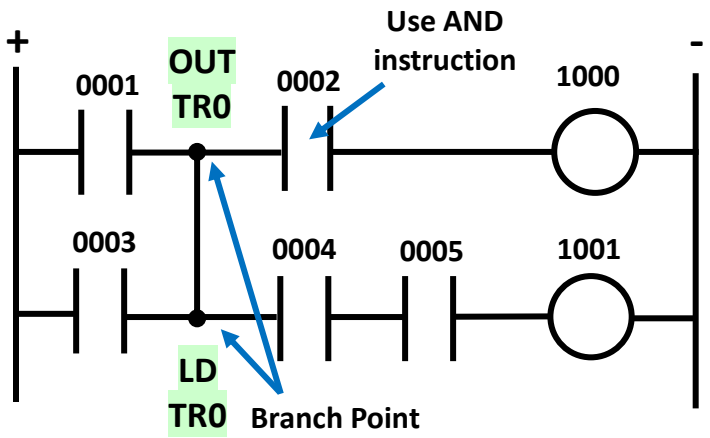


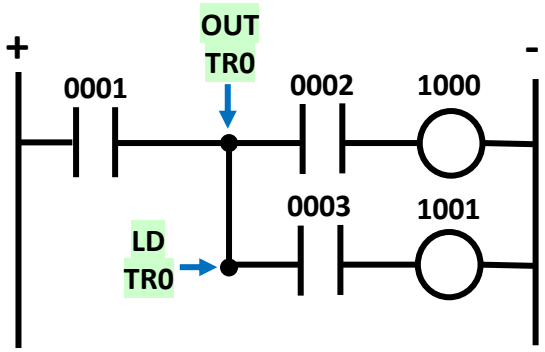
Figure 2.1 – Special branch point for the use of TR

- TR instruction can only be used with the OUT and LD instructions.
- The OUT TR instruction should be used at the first branch point.
- The LD TR instruction is used after the second row of the branch point.
- The first instruction after OUT TR or LD TR instruction must be connected with the AND instruction.

This is a simple example of using the TR instruction:

EXAMPLE 1

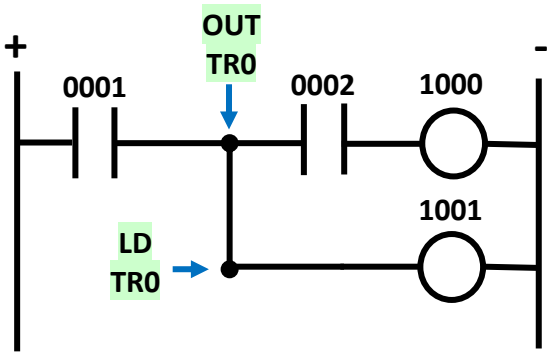
When a branch point **with a separate input** is connected to an output.



Address	Instruction	Data
0000	LD	0001
0001	OUT	TR0
0002	AND	0002
0003	OUT	1000
0004	LD	TR0
0005	AND	0003
0006	OUT	1001

EXAMPLE 2

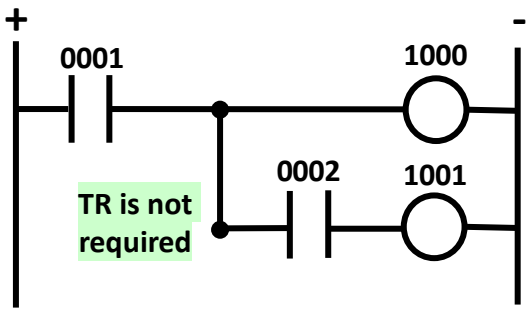
When a branch point **without a separate** input is connected to an output.



Address	Instruction	Data
0000	LD	0001
0001	OUT	TR0
0002	AND	0002
0003	OUT	1000
0004	LD	TR0
0006	OUT	1001

EXAMPLE 3

TR instruction is **not required** when there are no contact input after the branch point in the first row of the instruction block.



ADDRESS	INSTRUCTION	DATA
0000	LD	0001
0001	OUT	1000
0002	AND	0002
0003	OUT	1001

DESIGN OF CONTROL SYSTEMS USING PLC

STEP 1:
Determine the machine sequence of operation
By drawing flowchart to show the sequence of operation

STEP 3:
Writing of the program
Write the ladder diagram program by following the control system sequence of operation as determined by step one.

STEP 5:
Running the system
Test run the system thoroughly until it is safe to operate by anyone.

STEP 2:
Assignment of inputs and outputs
Identify the input and output device in the system

STEP 4:
Programming into PLC memory
Through console or computer by using CX programmer software

EXAMPLE OF CONTROL SYSTEMS DESIGN USING PLC

Design the following sequences:

A DC motor should operate continuously when the Start button, S1, is pressed and only turn off when the Stop button, S2, is pressed.

STEP 1: Determine the process sequence

The activity	The sequence
S1 ; ON	DC Motor, M1 ; ON
S2 ; ON	DC Motor, M1 ; OFF

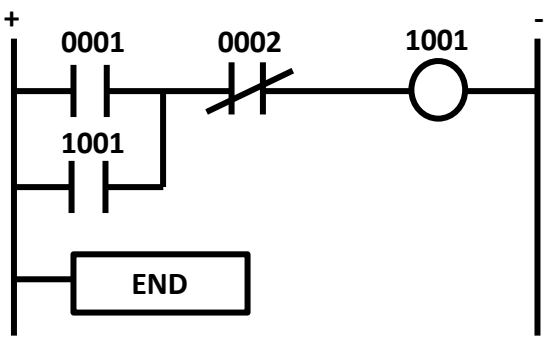
Can also use the flow chart as a reference

STEP 2: Assignment of inputs and outputs

Input Devices	Description
0001	S1
0002	S2

Output Devices	Description
1001	M1

STEP 3: Writing of the program



STEP 4 : Write Mnemonic Code if Programming into PLC by using Console

Address	Instruction	Data
0000	LD	0001
0001	OR	1001
0002	AND NOT	0002
0003	OUT	1001
0004	FUN (01)	

Ladder Diagram can be programmed by using CX programmer software (OMRON)

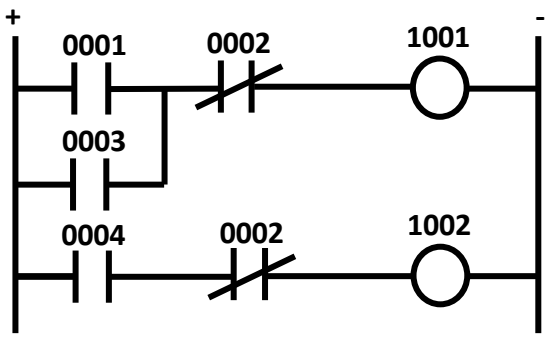
REFLECTION QUESTION

1. Explain the difference between the instruction AND and AND NOT instruction with the aid of a ladder diagram.

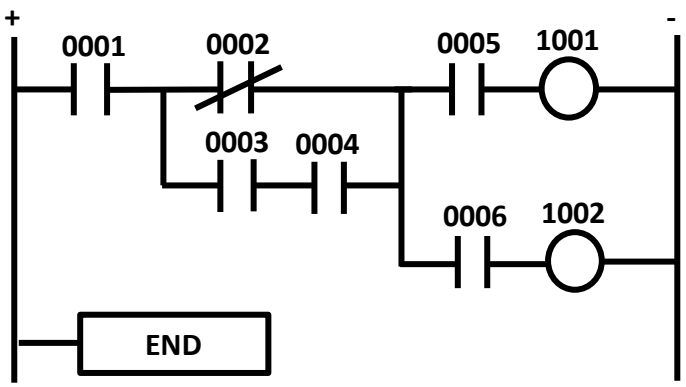
2. List the steps to design control system using PLC Programming.

3. Explain the function of AND LD and OR LD.

4. Explain the operation of the ladder diagram below.



5. Convert the ladder diagram below to mnemonic code.



6. Explain temporary relays (TR) and give one example of using the instruction?

You can check your answer via this link or QR code

https://cutt.ly/BCF5QDO

3.0 TIMER & COUNTER INSTRUCTION SET

Understand TIMER & COUNTER Instruction Set

1. Timer Instruction Set
2. Counter Instruction Set

TIMER INSTUCTION SET

A **timer (TIM)** is an internal PLC instruction used to delay time for input and output signals. The instructions that require numbers TIM (N) and the set value (SV) as shown in Figure 3.1.

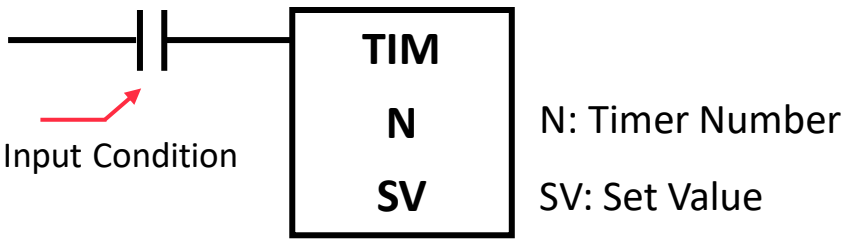
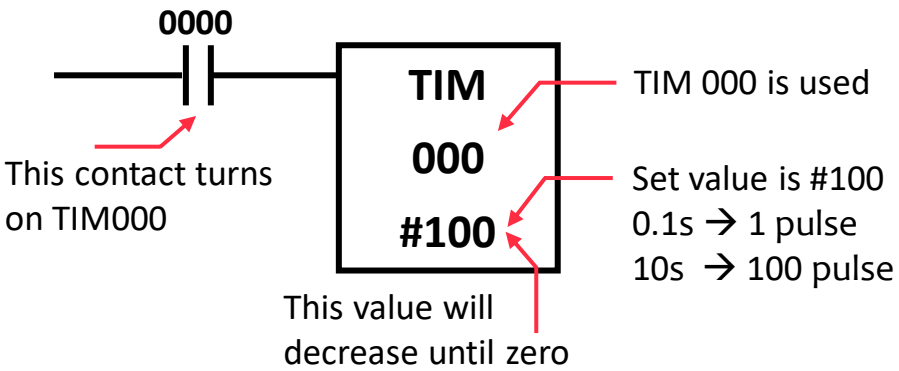


Figure 3.1 – Timer Instruction Set

The range of numbers (N) for TIM is from **000 to 255**, while the setting range of set values (SV) for the TIM is the BCD between **#0000** to **#9999.**

TIM operates a timer with 1 pulse scan time is 0.1s. For example, if timer be set to 5 seconds, then the set value (SV) is #0050.

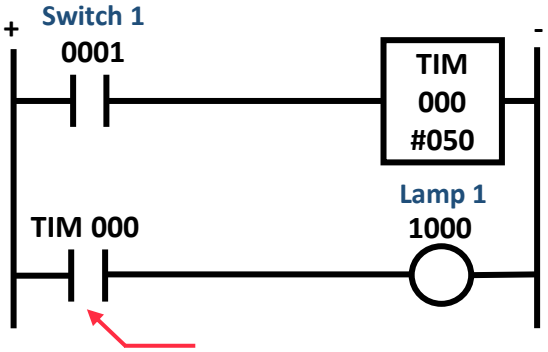


Timer is activated when the input condition is turns ON, and it will be reset to the set value (SV) when the input condition is OFF.

APPLICATION OF TIMER PROGRAM (ON DELAY)

Example

Lamp 1 will turn ON after 5 second when switch 1 is turned ON. Draw a ladder diagram and mnemonic code.

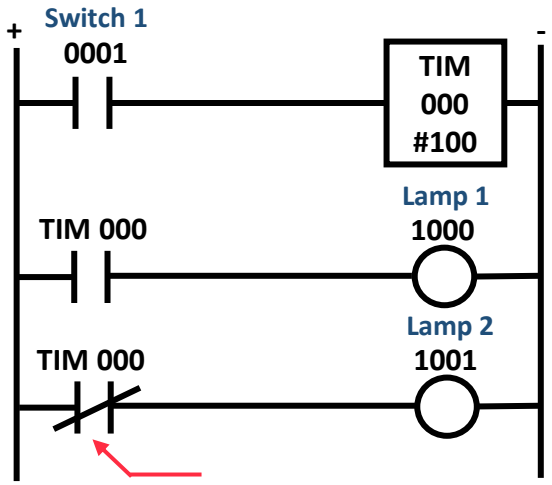


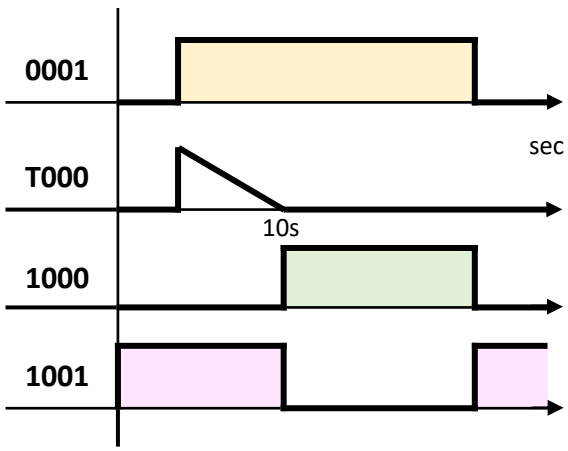
Address	Instruction	Data
0000	LD	0001
0001	TIM	000
		#050
0002	LD	TIM000
0003	OUT	1000

After 5 seconds contact TIM000 will change to NC and the lamp will turn ON.

Example 2

When switch 1 is turned ON, the TIM000 timer is triggered after 10 seconds. Then, lamp 1 will be turned ON and lamp 2 will be turned OFF. Draw a ladder diagram and timing diagram.





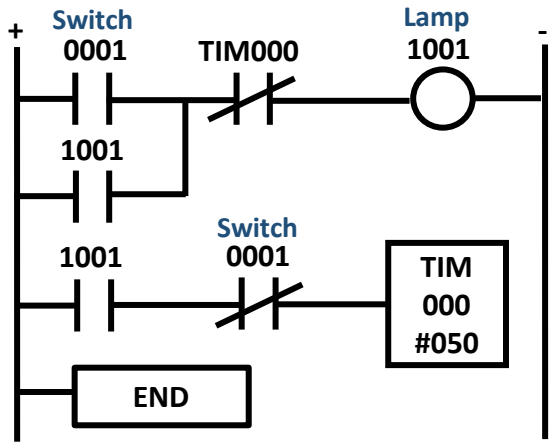
After 10 seconds contact TIM000 will change to NO and lamp 2 will turn OFF.

APPLICATION OF TIMER PROGRAM (OFF DELAY)

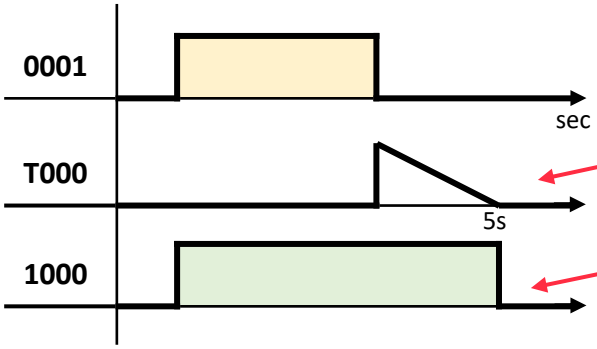
The OFF delay timer circuit can be implemented using ON delay timer circuit.

Example

When switch is turned ON, the lamp will turn ON. As long as switch is ON, the timer will not be active. When switch is turned OFF, the timer will start to trigger. After 5 seconds, the lamp will turn OFF.



Address	Instruction	Data
0000	LD	0001
0001	OR	1001
0002	AND NOT	TIM000
0003	OUT	1001
0004	LD	1001
0005	AND NOT	0001
0006	TIM	000
		#050
0007	FUN (01)	



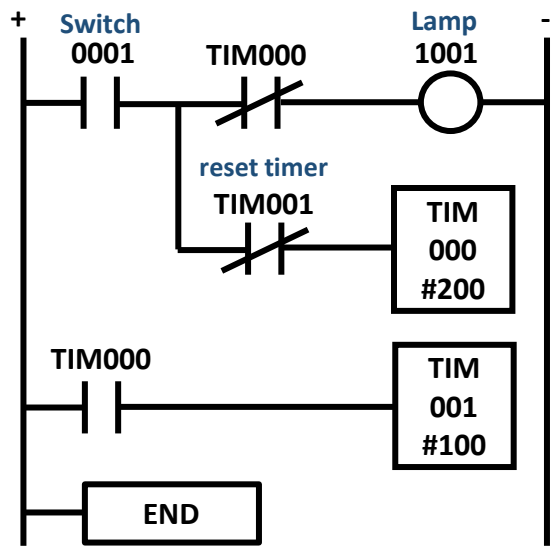
The timer will start to trigger if the switch is turned OFF.

The lamp will only turn OFF after a delay of time.

ON delay timer is a timer that is triggered when the input switch is turned ON. While OFF delay timer is a timer that is triggered when the input switch is turned OFF. There is a delay before the output is turned ON or OFF.

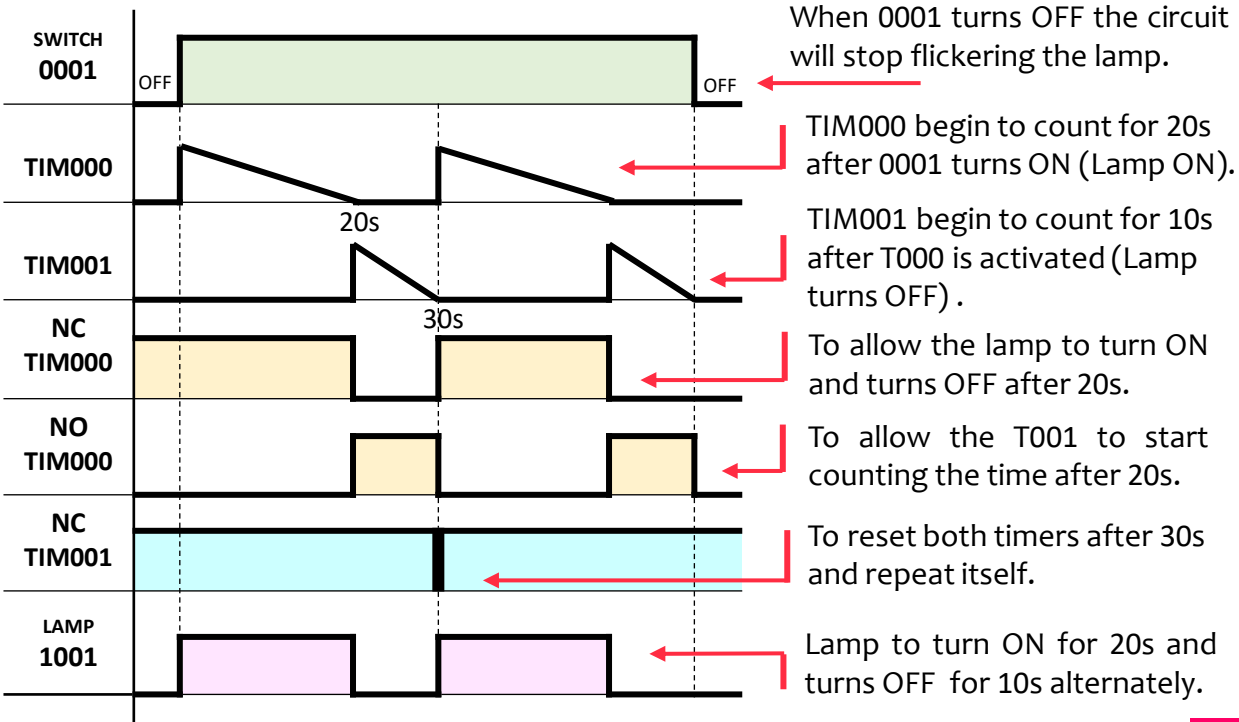
LIGHT FLICKER CIRCUIT USING TIMER PROGRAM

A light flicker circuit is a circuit that causes the lights to turn ON and OFF alternately. In this program, two timers are required. One is used to determine the ON time, while the other one is for the OFF time.



Address	Instruction	Data
0000	LD	0001
0001	OUT	TR0
0002	AND NOT	TIM000
0003	OUT	1001
0004	LD	TR0
0005	AND NOT	TIM001
0006	TIM	000
		#200
0007	LD	TIM000
0008	TIM	001
		#100
0009	FUN (01)	

Timing diagram illustrates a system's input/output status.



COUNTER INSTUCTION SET

A **counter (CNT)** is an internal PLC instruction used to count pulses. The instructions that require numbers CNT (N) and the set value (SV). The counter have two input condition with a clock pulse (CP) and reset input (R) as shown in Figure 3.2.

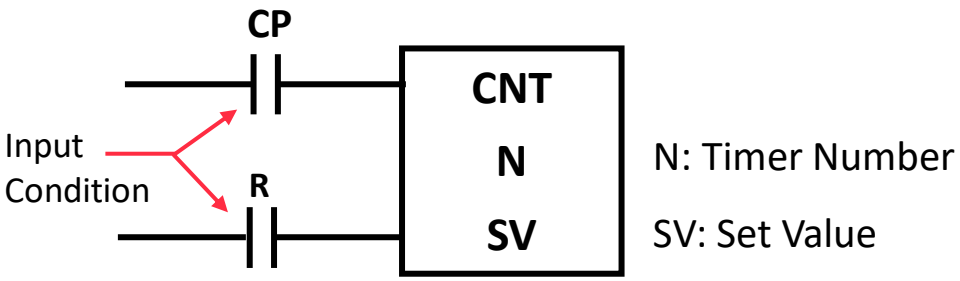
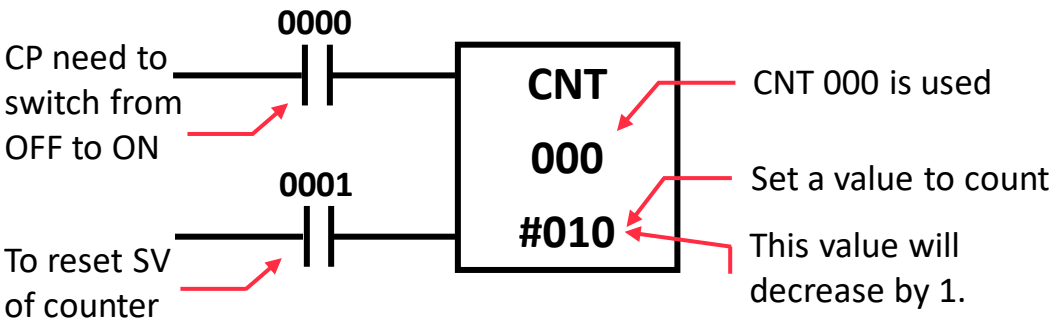


Figure 3.2 – Counter Instruction Set

The range of numbers (N) for CNT is from **000 to 255**, while the setting range of set values (SV) for the CNT is the BCD between **#0000** to **#9999.** Counter and timer cannot use the same range of numbers (N) in a program.

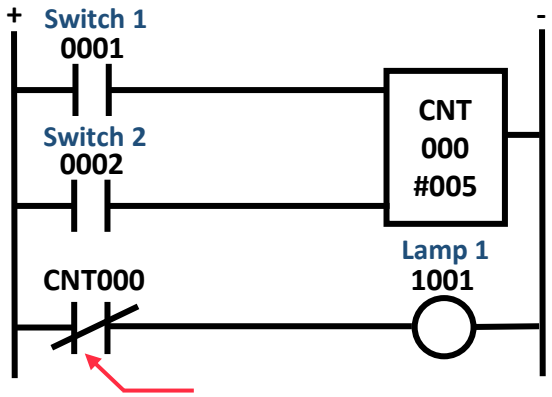


Each time input CP (0000) is switched from OFF to ON, the counter's set value decreases by one. Once it reaches zero, the counter will operate. The counter will reset if input R (0001) is ON. The counter can not count the transition condition at CP when R is ON. The counter can start to count the input at CP after R is turned OFF.

APPLICATION OF COUNTER PROGRAM

Example 1

Lamp 1 will turn OFF after Switch 1 is turned ON by 5 times. Switch 2 will reset the counter and lamp will turn ON again. Draw a ladder diagram and mnemonic code.

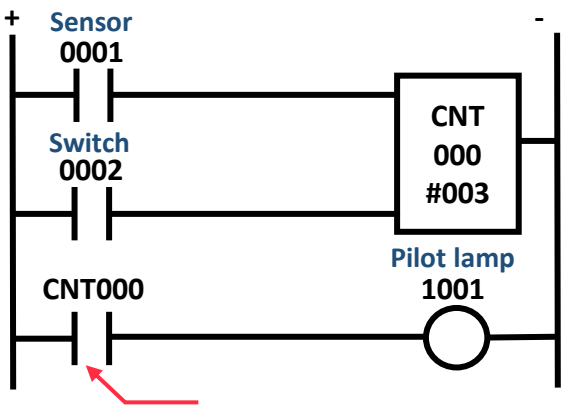


Address	Instruction	Data
0000	LD	0001
0001	LD	0002
0003	CNT	000
		#005
0004	LD NOT	CNT000
0005	OUT	1001

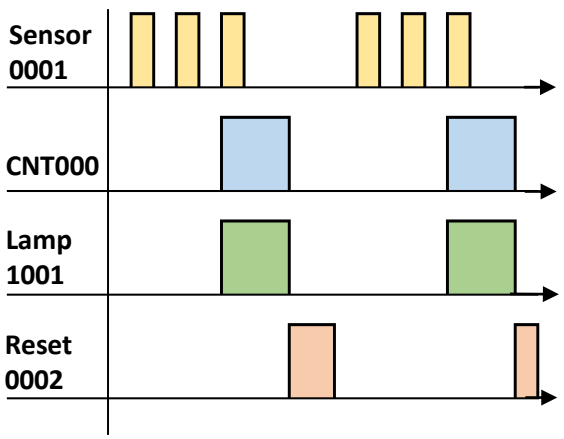
Contact CNT000 will change to NO and the lamp 1 will turn OFF after the counter has been counted 5 times.

Example 2

Draw a timing diagram and ladder diagram that will activate a pilot lamp after a sensor detects the presence of 3 products. The pilot lamp will deactivate after switch reset is pressed.



Contact CNT000 will change to NC and the pilot lamp will turn ON after the counter has been counted 3 times.

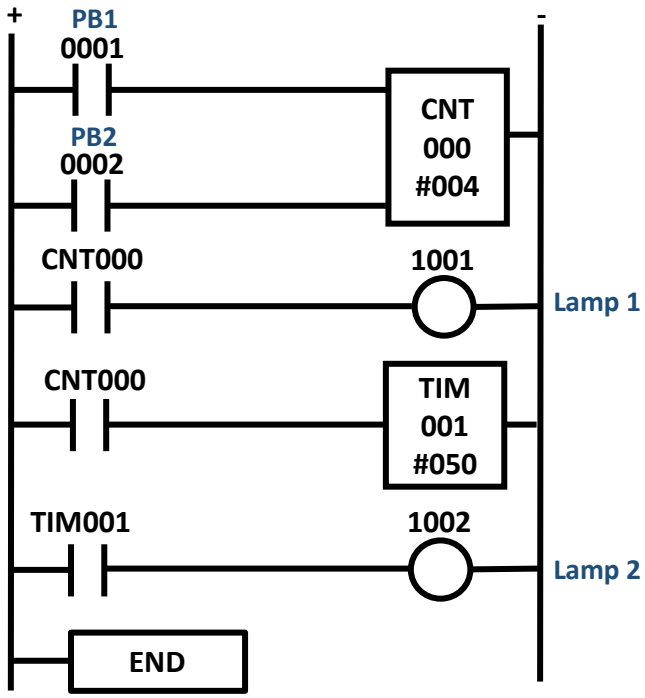


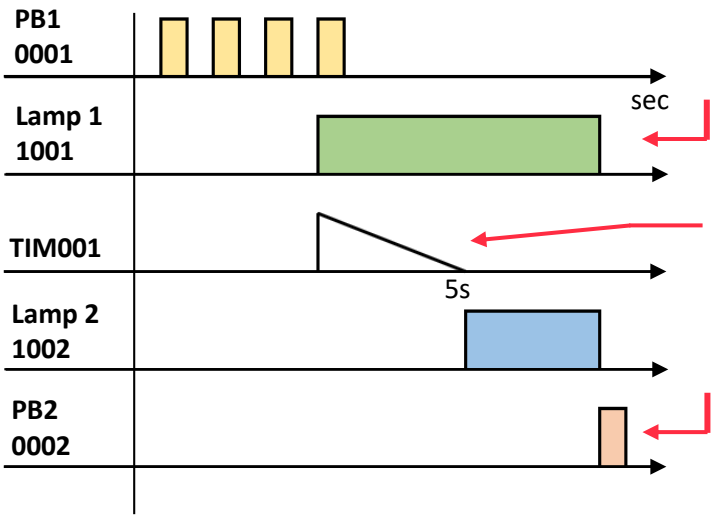
Timing Diagram

SIMPLE APPLICATION USING TIMER AND COUNTER

Example

When push button 1 (PB1) is pressed for 4 times, the Lamp 1 will be ON. When Lamp 1 is ON, TIM001 will be activated and, after 5 seconds, Lamp 2 will be ON. Push Button 2 (PB2) is used to reset the counter. Draw a ladder diagram and timing diagram for the system.





After PB1 has been activated four times, CNT000 will be triggered, and Lamp 1 will be turn ON.

At the same time, TIM001 begins to count for 5 seconds, and after TIM001 is activated, Lamp 2 turns ON.

Lamp 1 and lamp 2 will remain ON until PB2 is pressed.

REFLECTION QUESTION

1. State the range of numbers that can be used to identify a timer (TIM) and counter (CNT).

2. Draw a ladder diagram that a pilot lamp will turn ON after 10 seconds when switch 1 is turned ON.

3. What is the difference between a ON delay and a OFF delay timer circuit?

4. Explain the counter instruction set in PLC programming using appropriate diagrams.

5. Draw a ladder diagram that will turn ON a motor after a toggle switch is closed. The motor should stop automatically after two minutes, while the toggle switch is still closed.

6. Determine the SV values of timer and counter to have a delay time of 6 hours.

7. A pilot lamp will turn ON after sensor detects a presence of 12 objects. The system can be reset after push button is pressed or after a time delay of one minutes. Draw a ladder diagram and write out the mnemonic code for this operation.

4.0 SPECIAL INSTRUCTION SET

Understand Special PLC Instruction Set

1. HOLDING CIRCUIT
2. SET/RESET
3. KEEP
4. DIFU / DIFD
5. INTERLOCK / INTERLOCK CLEAR
6. JUMP / JUMP END

To understand the special instructions of the PLC program, firstly we must know about the internal relay (IR).

INTERNAL RELAY (IR)

Internal Relay (IR) is one bit memory device. If the program is used with IR, this bit stores the data and maintains an ON or OFF status regardless of whether the input **set** is ON or OFF.

IR is an imaginary programmed relay that has coils and contacts but is not connected to a real output (relay). So, internal relay **cannot** have addresses that start with the data '10'.

For the OMRON PLC CPM2A, the bit data for IR addresses are:

20000 to 23115 (928 bits)

IR can be used in other special instruction sets such as KEEP, SET and RSET, DIFU, DIFD. These instructions are used to control ON and OFF state output bits in the IR. Figure 4.1 shows the ladder diagram using holding circuit with internal relay (IR).

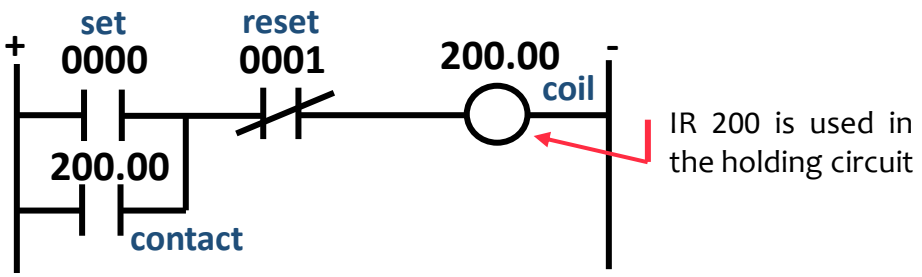


Figure 4.1 – Internal Relay (IR)

All PLCs have IR, however the internal numbering scheme work bit for them depends on the PLC model.

HOLDING CIRCUIT (LATCH)

Holding Circuit is used as a latch. It is used to keep the status of the specified bit based on two input conditions. These two input conditions are indicated by the symbols **S** and **R** as shown in Figure 4.2. S is the set input and R is the reset input.

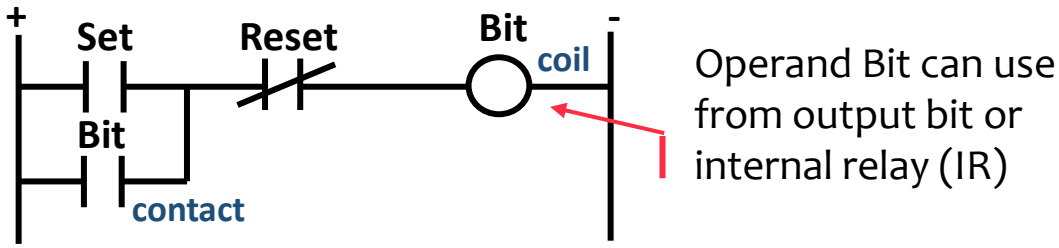
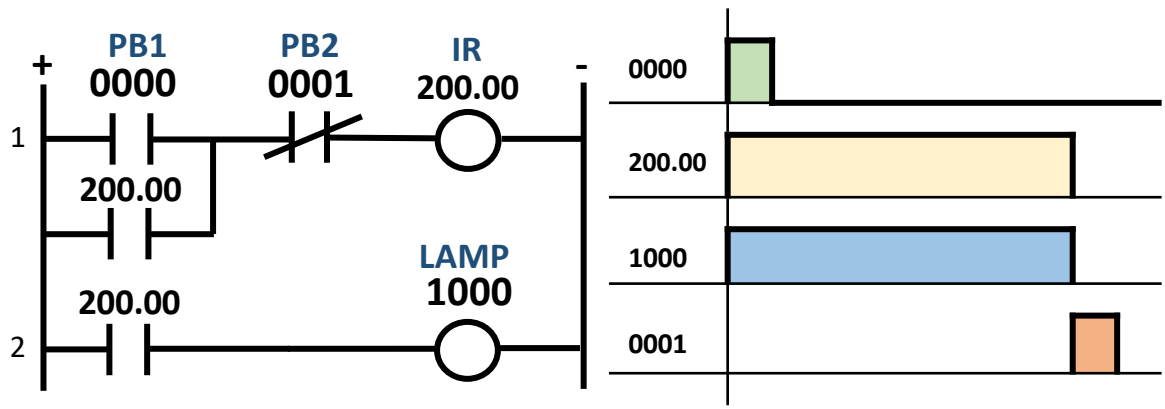


Figure 4.2 – Holding Circuit

APPLICATION OF HOLDING CIRCUIT



When push button 1 (0000) is momentarily pressed (ON then OFF), output IR 200 is energized. Because of IR 200 output is held by a latching, it will remain ON even if 0000 is turned OFF. This condition will also cause the output Lamp (1000) to remain turned ON.

When push button 2 (0001) is pressed, the internal relay of the IR 200 is deenergized and the IR 200 is turned OFF. As a result, the lamp (1000) will also be turned OFF.

EXAMPLE 1

When the start button is pressed and released, a motor will rotate, and the conveyor will move to bring out the product. The sensor will count 12 products and inserting it into the box as shown in Figure 4.3. Then, the motor will be stopped the process.

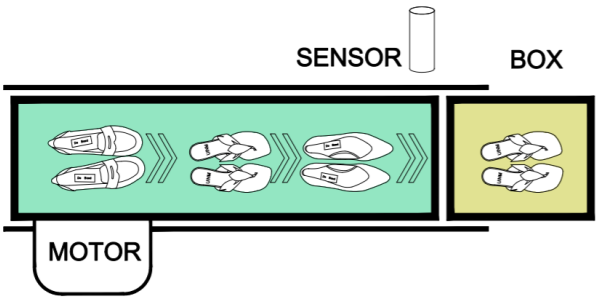
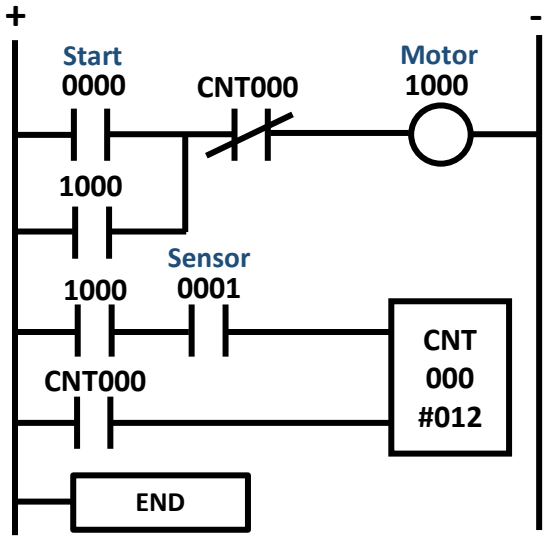


Figure 4.3



EXAMPLE 2

Design PLC ladder diagram by using Holding Circuit to implement the timing diagram shown in Figure 4.4:

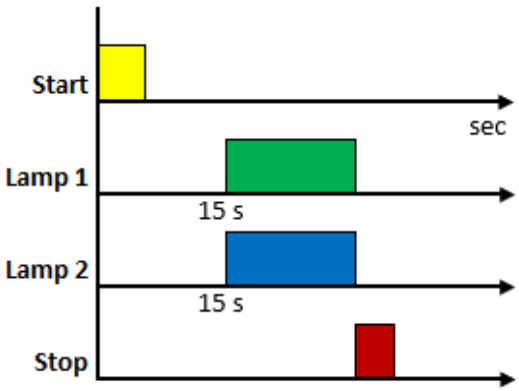
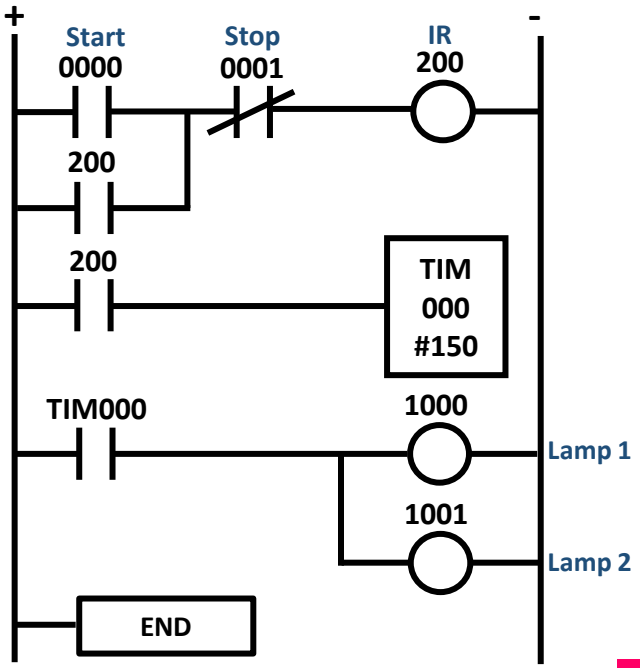


Figure 4.4



EXAMPLE 3

Design PLC ladder diagram by using Holding Circuit to implement the timing diagram as shown in Figure 4.5:

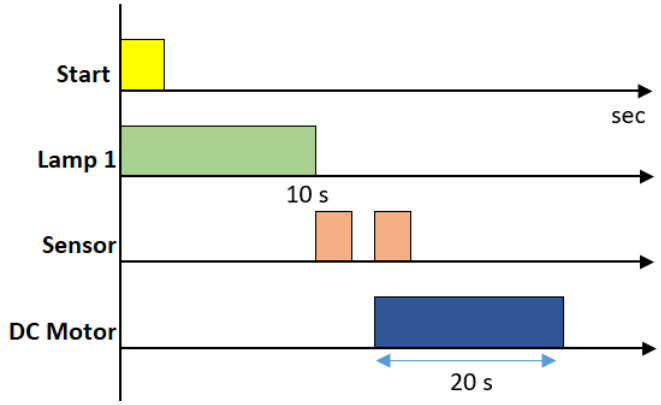
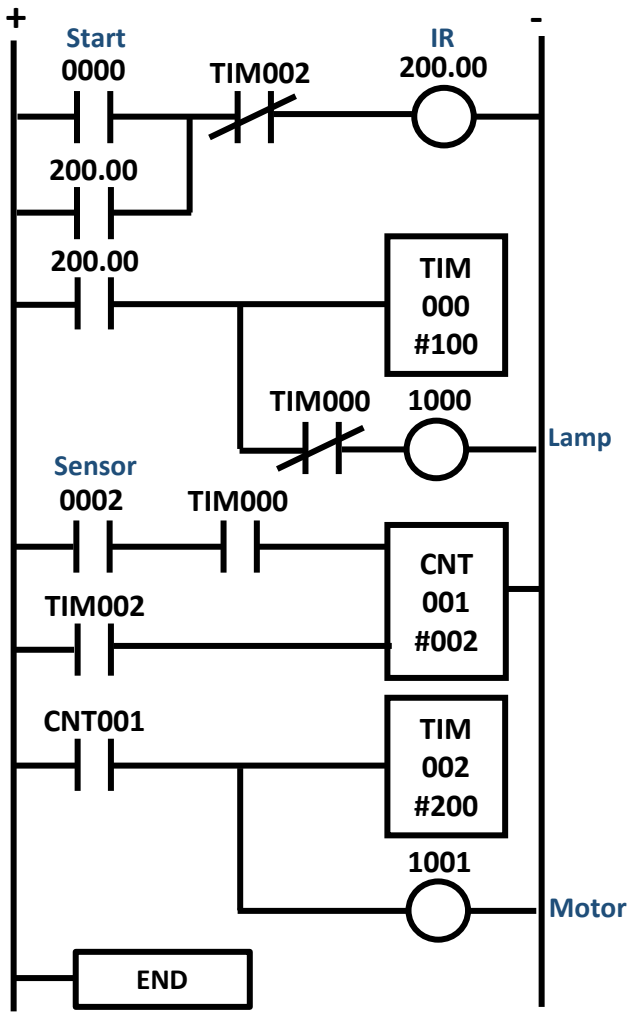


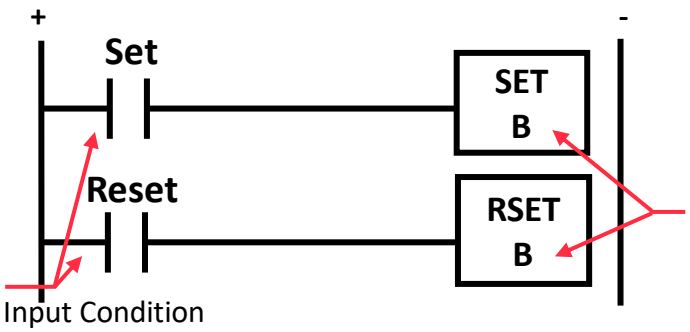
Figure 4.5



Address	Instruction	Data
0000	LD	0000
0001	OR	200.00
0002	AND NOT	TIM002
0003	OUT	200.00
0004	LD	200.00
0005	TIM	000
		#100
0006	AND NOT	TIM000
0007	OUT	1000
0008	LD	0002
0009	AND	TIM000
0010	LD	TIM002
0011	CNT	001
		#002
0012	LD	CNT001
0013	TIM	002
		#200
0014	OUT	1001
0015	FUN(01)	

SET/RESET

SET and RESET are optional instructions that can be used to keep the output. The SET instruction will turn ON the operand bit (B) when the input condition is ON, and it stays ON even when the input condition is OFF. RSET will turn OFF the operand bit when the input condition is ON. Figure 4.6 shows the Set and Reset instruction using ladder diagram program.



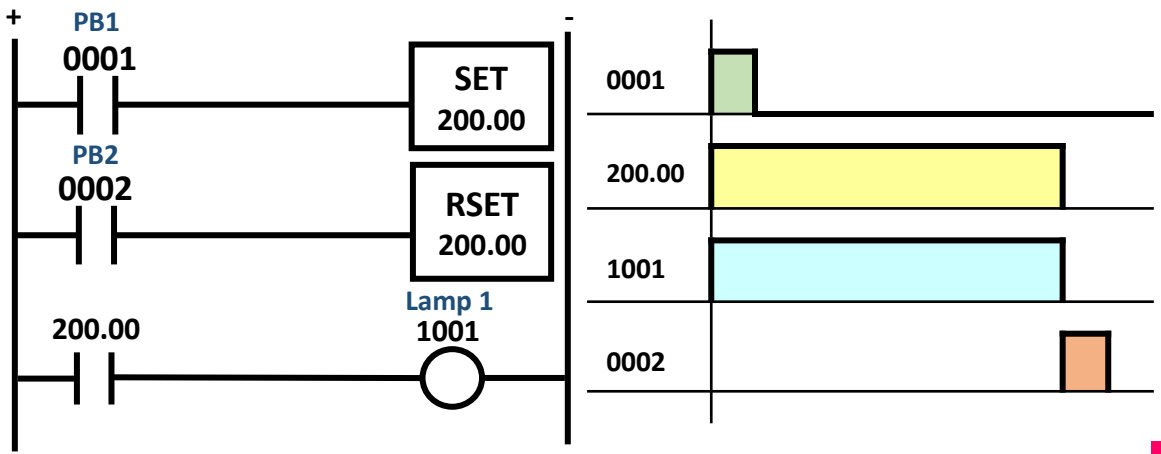
Bit operand (B) may consist of data output or internal relay (IR)

Set and Reset must used a same data

Figure 4.6 – SET and RESET instruction

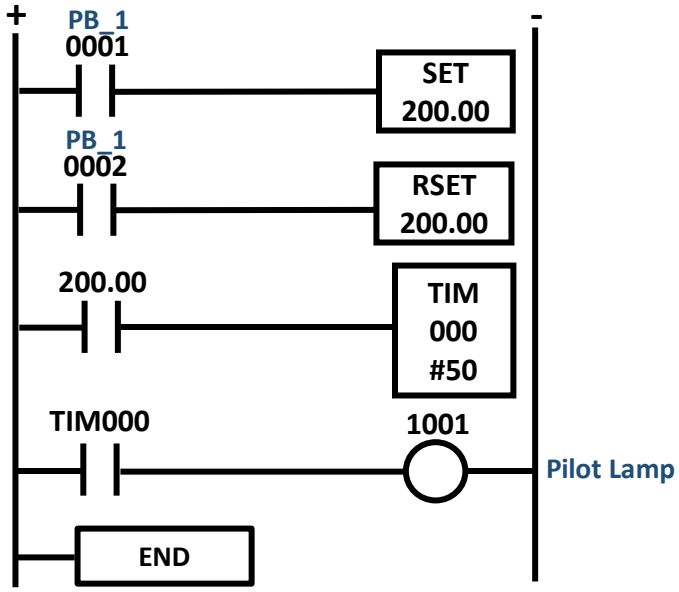
APPLICATION OF SET/RESET

When PB1 (0001) is momentarily pressed, SET IR 200.00 is energized and it will remain ON even if PB1 is turned OFF. This condition keeps the output Lamp 1 (1001) is turned ON. When PB2 (0002) is pressed, the RSET IR 200.00 is turned OFF. As a result, the lamp (1001) will also be turned OFF.



Example 1

By using the SET/RESET instruction, create a ladder diagram that will light up a pilot lamp after 5 second the push button start (PB1) is pressed. Then, after pushing a push button stop (PB2), the lights will be turned off.



Example 2

Design PLC ladder diagram by using SET and RESET instruction to implement the timing diagram as shown in Figure 4.7:

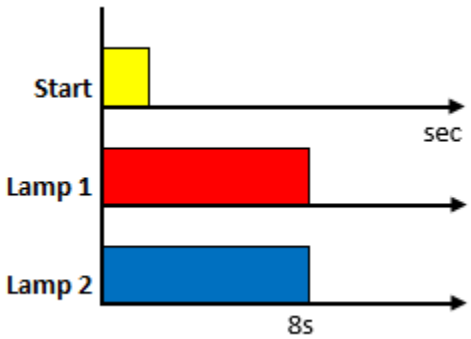
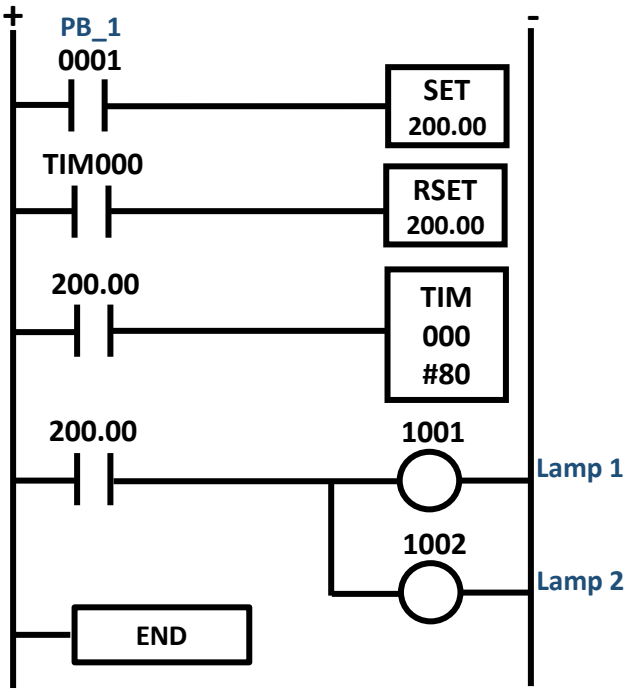


Figure 4.7



KEEP

The KEEP instruction is another optional instruction that can also be used to maintain the output. KEEP instruction is used to keep a status bit operation based on two input conditions.

These two input conditions are indicated by set (S) and reset (R) as shown in Figure 4.8. Bit operand (B) may consist of data output or internal relay (IR).

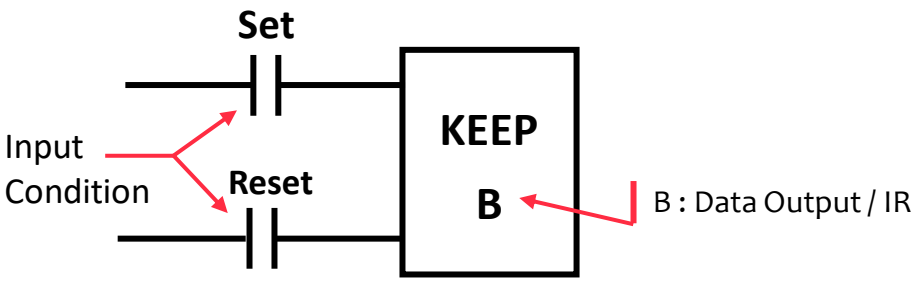
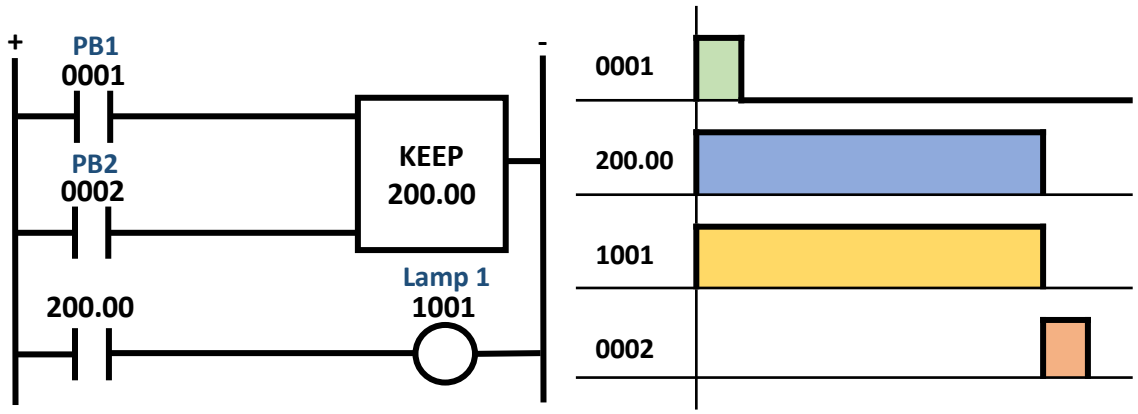


Figure 4.8 – KEEP instruction

When S is in the ON state, the operation of the KEEP instruction will be ON and remain ON until reset, regardless of whether Set is ON or OFF. When Reset is in the ON state, the operation of KEEP instruction is OFF.

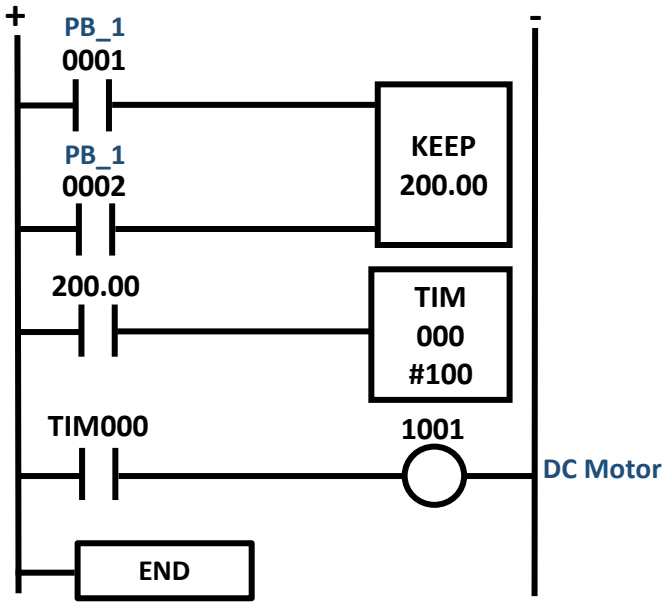
APPLICATION OF KEEP INSTRUCTION

KEEP functions in almost the same way as a holding circuit and the SET/RSET instruction.



Example 1

Design a ladder diagram by using the KEEP instruction. The DC motor will run automatically after 10 second the push button start (PB1) is pressed. Then, after pushing a push button stop (PB2), the DC motor will stop.



Example 2

Design PLC ladder diagram by using KEEP instruction to implement the timing diagram as shown in Figure 4.9:

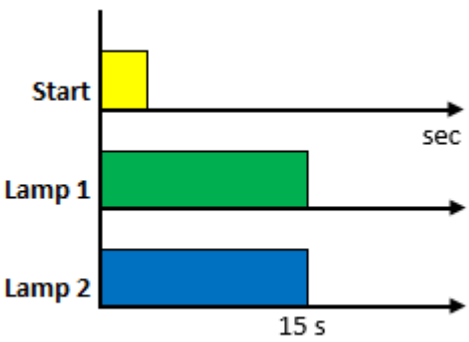
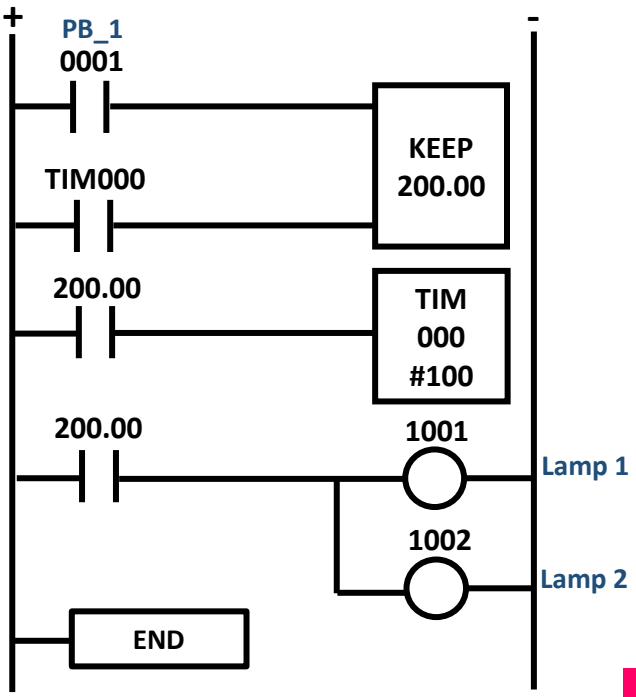


Figure 4.9



EXAMPLE PROGRAM : HOLDING CIRCUIT AND KEEP

Refer to the figure 4.10, it shows the system connection to control the traffic light. When start push button (PB1) is pressed, red light will turn ON for 5 seconds. After that, green light will turn ON for 10 seconds and red light will turn OFF. The operation will repeat simultaneously until stop push button (PB2) is pressed to stop the operation. Construct this sequence using PLC ladder diagram.

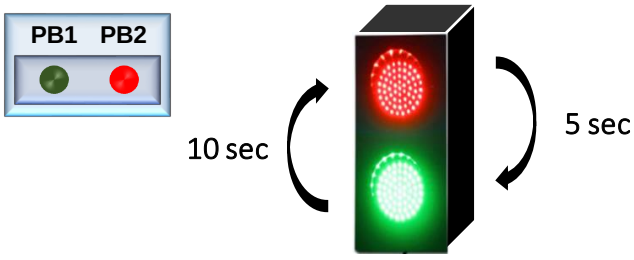
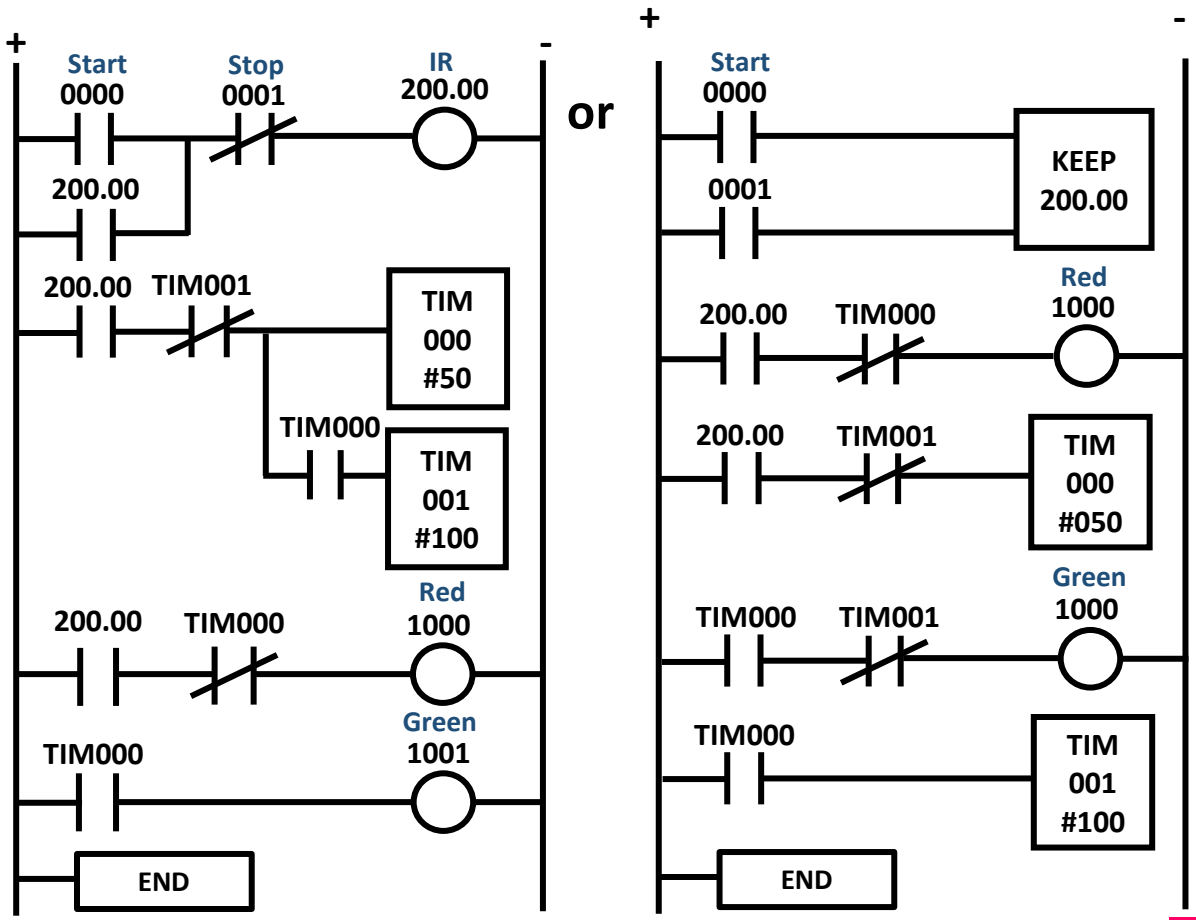


Figure 4.10

The ladder diagram below shows two types of programming design that can be used in this situation.



EXAMPLE PROGRAM : KEEP AND SET/RSET

In the final round of the polytechnic quiz competition, two students from the different polytechnics will compete against each other as shown in figure 4.11. If one of them presses the button faster, the desk light will remain on for 10 seconds until turning off. Those who press the button too late, their lights will not turn on. Draw the ladder diagram by using KEEP and SET / RSET instruction.

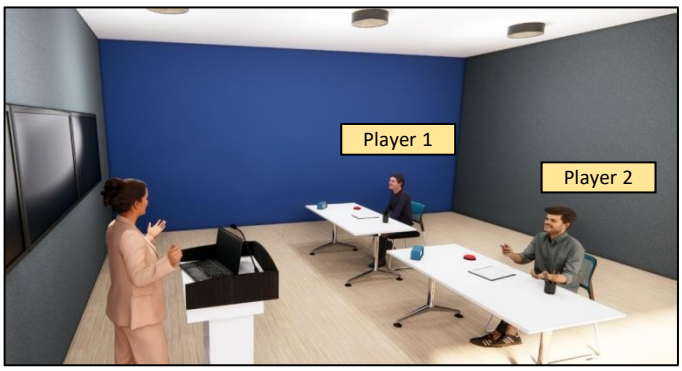
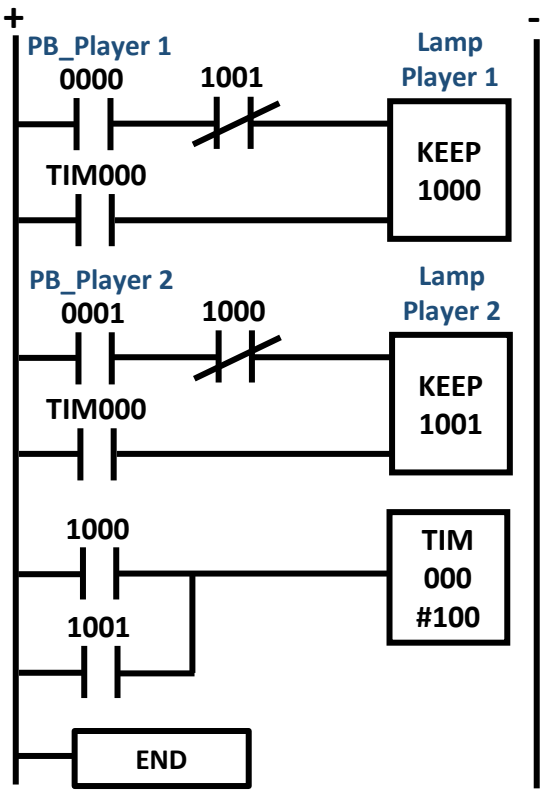


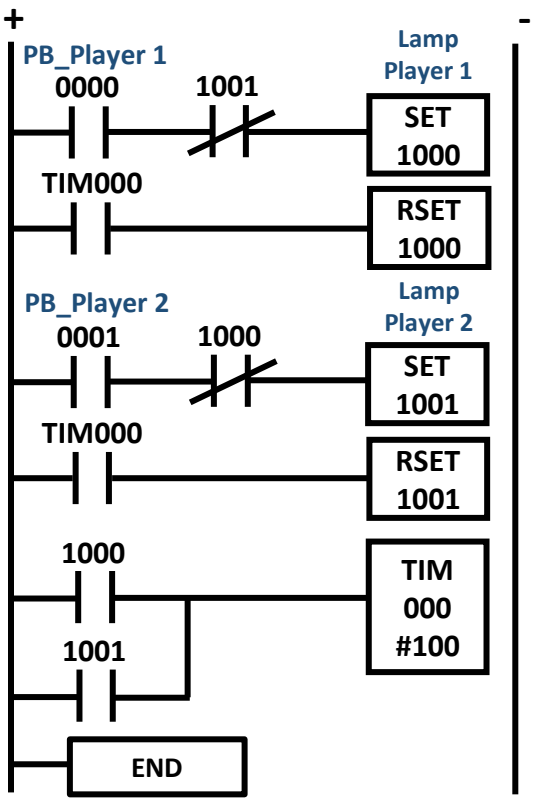
Figure 4.11

KEEP instruction



or

SET/RESET instruction



DIFFERENTIATE UP (DIFU) AND DIFFERENTIATE DOWN (DIFD)

The Differentiate UP (DIFU) and Differentiate Down (DIFD) instruction are a special instruction that will turn ON the output in a very short time.

DIFU instructions will turn ON the bit when the input condition changes from **OFF to ON** (rising edge). When the input is turned ON, it is used to perform operations that require only one scan cycle. Figure 4.12 shows the ladder diagram and timing diagram for DIFU instruction.

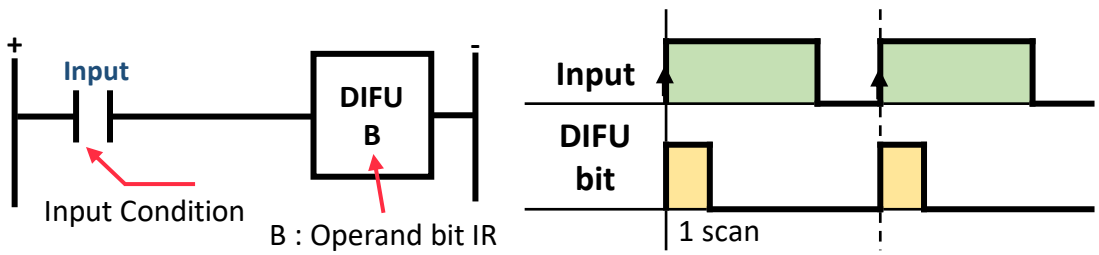


Figure 4.12 - DIFU instruction

DIFD instructions will turn ON the bit when the input condition changes from **ON to OFF** (falling edge). When the signal is turned OFF, it is used to perform operations that require only one scan cycle. Figure 4.13 shows the ladder diagram and timing diagram for DIFD instruction.

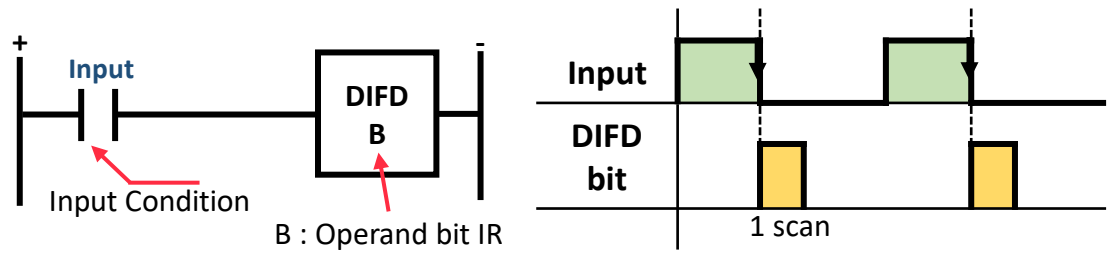
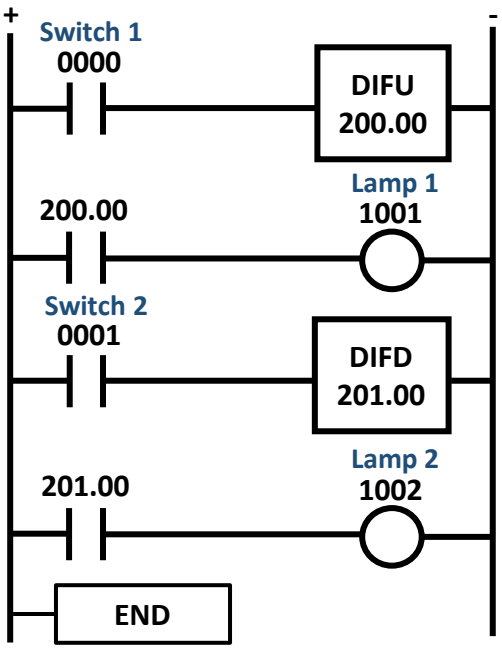
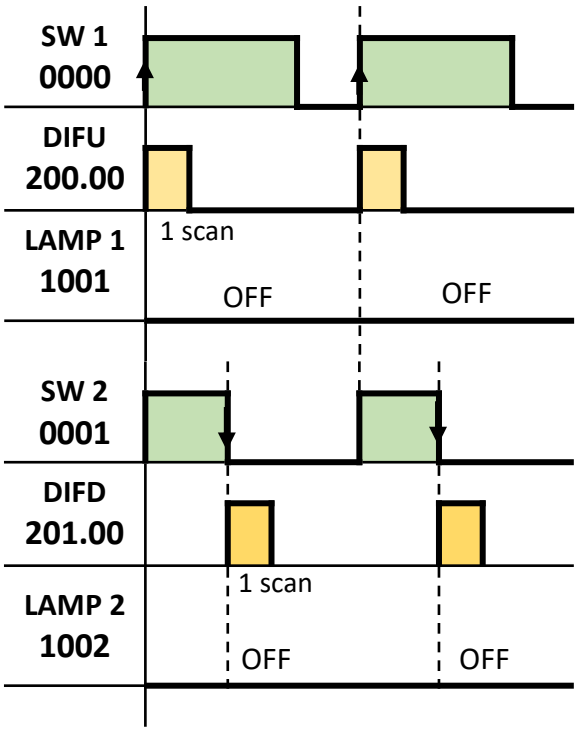


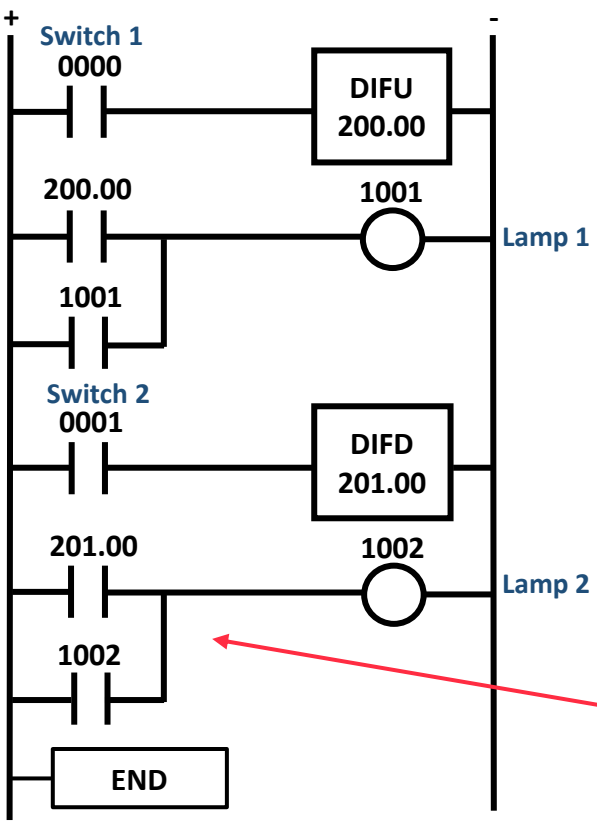
Figure 4.13 – DIFD instruction

APPLICATION EXAMPLE OF DIFU & DIFD





Based on the timing above, it shows the process output cannot be seen because the lamp will be ON **within a very short time and then OFF.**



Address	Instruction	Data
0000	LD	0000
0001	DIFU	200.00
0002	LD	200.00
0003	OR	1001
0004	OUT	1001
0005	LD	0001
0006	DIFD	201.00
0007	LD	201.00
0008	OR	1002
0009	OUT	1002

To solve the above problem, use the holding circuit concept to ensure that both lamps are always turned ON.

APPLICATION EXAMPLE OF DIFU

When the push button is pressed and released, motor will turn ON. Then, conveyor will move to carry out the box and stop after a limit switch detects the next box as shown in Figure 4.14. This process will be repeated by an operator in verify the quality of the product in the box. Draw a ladder diagram for the system.

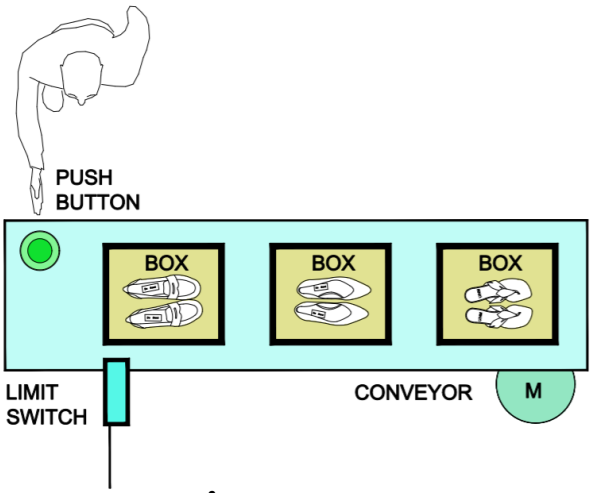
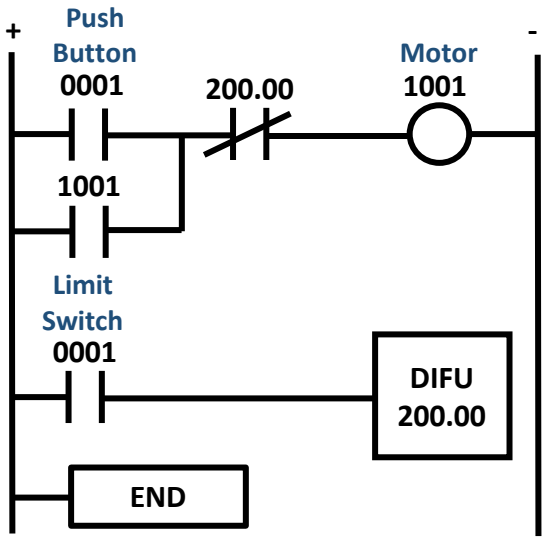


Figure 4.14



APPLICATION EXAMPLE OF DIFD

Design PLC ladder diagram by using DIFD to implement the timing diagram shown in Figure 4.15:

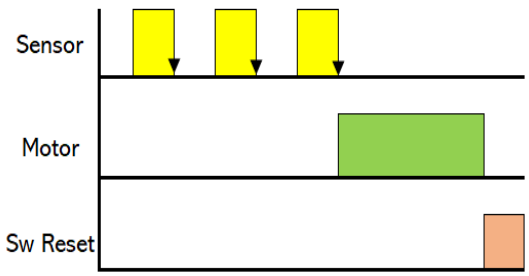
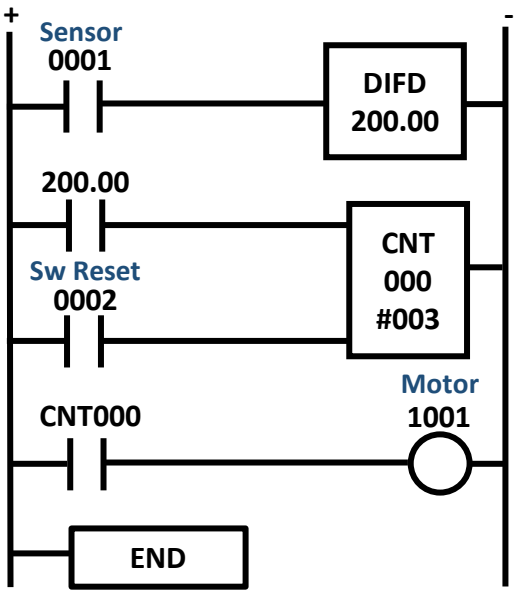


Figure 4.15



INTERLOCK / INTERLOCK CLEAR

The interlock (IL) and interlock clear (ILC) instructions are used to lock or skip all output conditions between IL and ILC when the input condition for IL is turned OFF. If IL is turned ON, no lock and skipping occur. IL and ILC must be used together as shown in Figure 4.16.

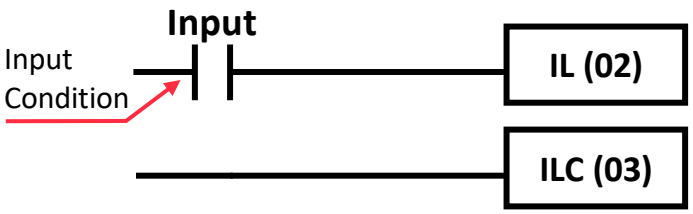
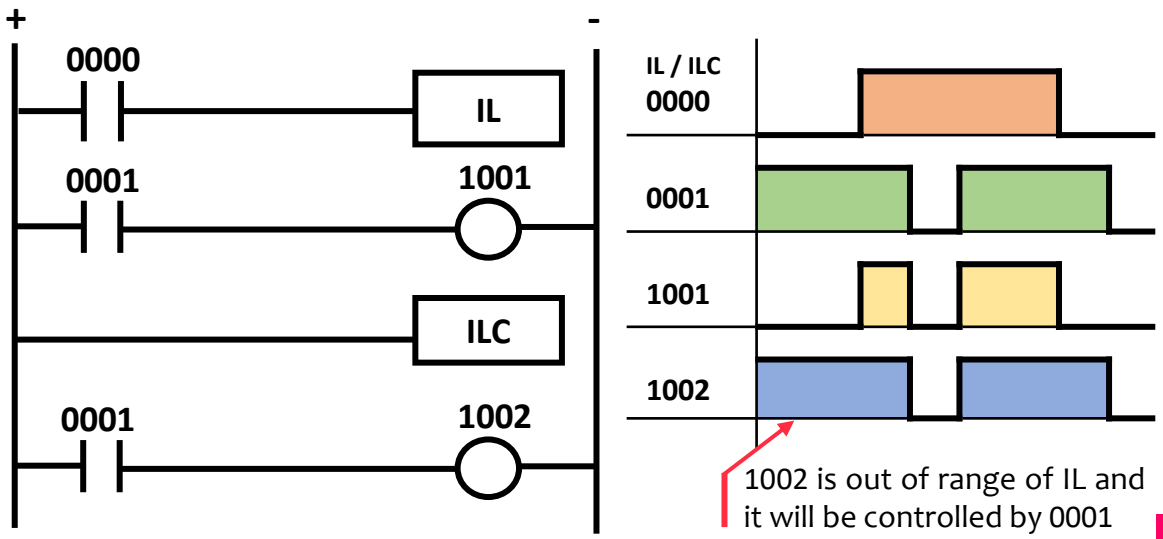


Figure 4.16 – Interlock & Interlock Clear Instruction

APPLICATION OF IL AND ILC

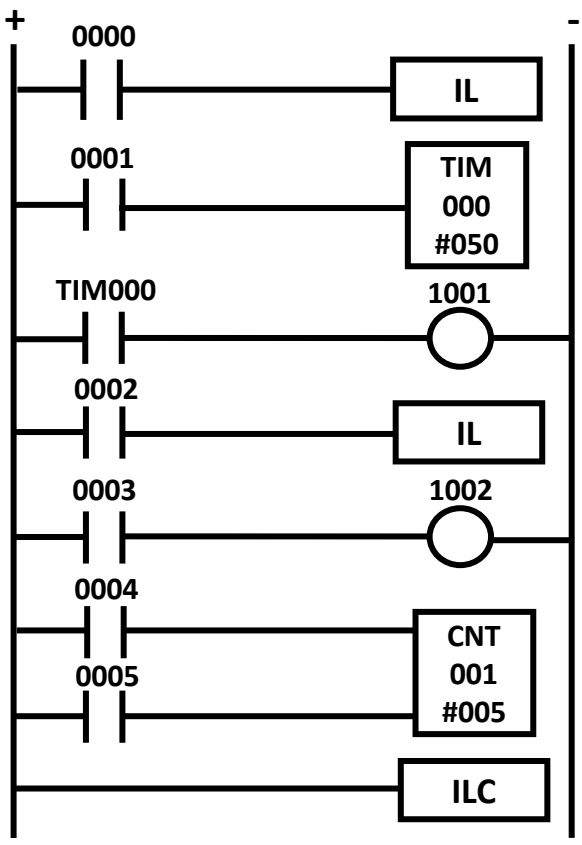
When input 0000 is turned OFF, output 1001 that is between IL and ILC will be skipped and will not function. Then, when input 0001 is turned ON only output 1002 it will be turned ON because it is outside the interlock range.

Now, when input 0000 is turned ON, all outputs that are between IL and ILC can execute normally, so output 1001 will be turned ON when switch 0001 is turned ON. Next, when input 0000 is turned OFF again, output 1001 will be turned OFF or reset.



INTERLOCK / INTERLOCK CLEAR

IL may be used many times in a row in the programme and should be ended by ILC. ILCs should not be used unless at least one IL is provided. The ladder diagram and mnemonic code in Figure 4.17 represents IL being used twice with one ILC.



Address	Instruction	Data
0000	LD	0000
0001	IL	
0002	LD	0001
0003	TIM	000
		#050
0004	LD	TIM000
0005	OUT	1001
0006	LD	0002
0007	IL	
0008	LD	0003
0009	OUT	1002
0010	LD	0004
0011	LD	0005
0012	CNT	001
		#005
0013	ILC	

Figure 4.17 – Application of IL being used twice with one ILC.

When input 0000 for 1st IL is turned ON and 0002 for 2nd IL is turned OFF, TIM000 will be activated for 5 seconds and output 1001 will be turned ON but CNT001 will not be changed and output 1002 will be turned OFF.

When 0000 and 0002 for both ILs are ON, the program between the 1st IL and the ILC will execute normally.

Next, when both of ILs are turned OFF again, TIM000 will be reset and outputs 1001 and 1002 will be turned OFF but CNT001 will not change. CNT001 will not reset when input 0005 is turned ON.

JUMP / JUMP END

The Jump (JMP) and Jump End (JME) instructions can be used to skip all output conditions between JMP and JME when the input condition for JMP is turned OFF. If JMP is turned ON all program can execute normally. Although this instruction is very similar to Interlock (IL), but the JMP instruction will retain or not changed the status of all its outputs after JMP is turned OFF. Figure 4.18 shows the JMP and JME instruction.

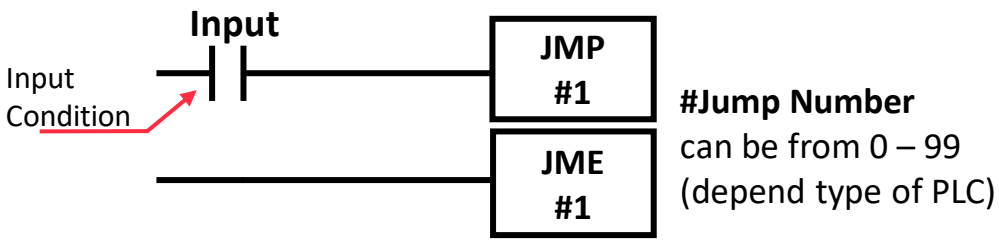
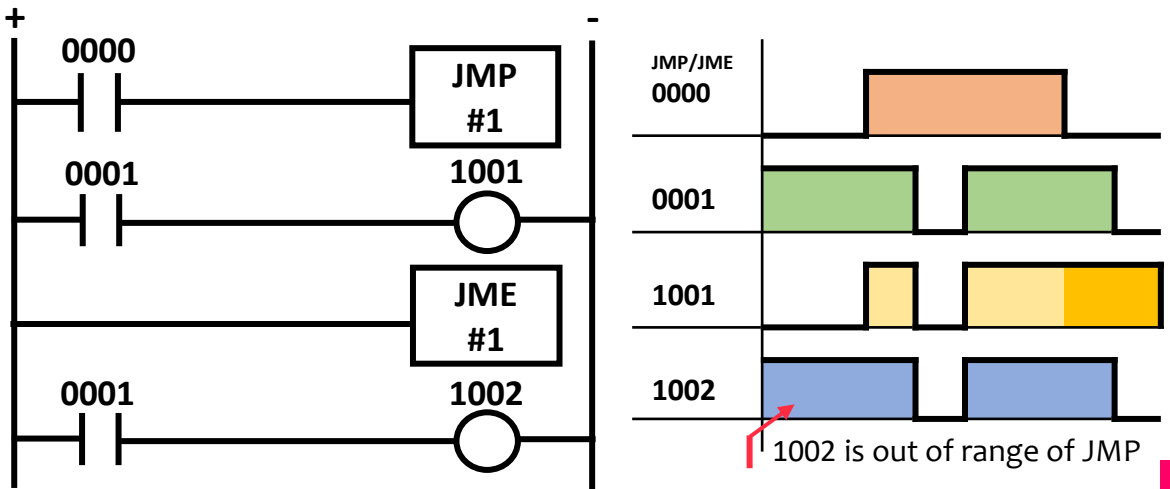


Figure 4.18 – Jump and Jump End Instruction

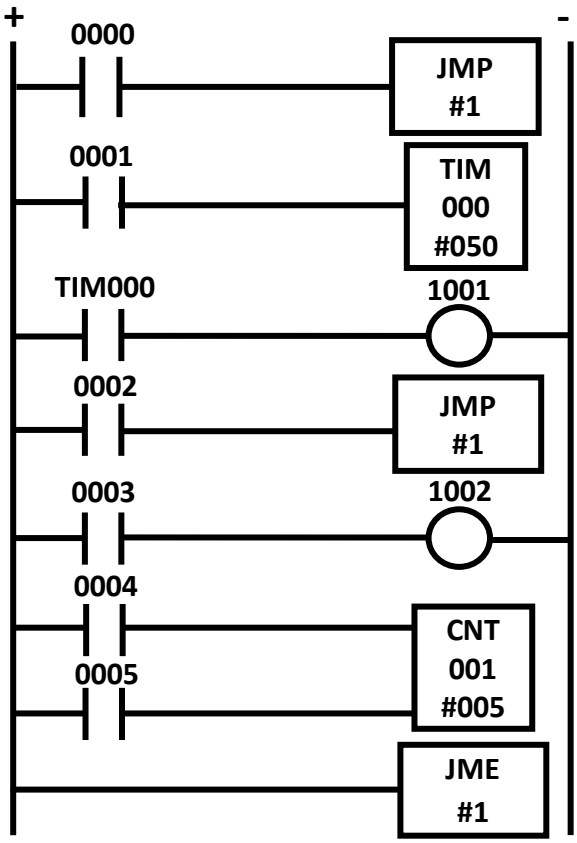
APPLICATION OF JMP AND JME

When input 0000 is turned OFF, output 1001 that is between JMP and JME will be skipped. Then, when input 0001 is turned ON only output 1002 it will be turned ON because it is outside the JMP range. Now, when input 0000 is turned ON, all outputs that are between JMP and JME can execute normally, so output 1001 will be turned ON when switch 0001 is turned ON. Next, when input 0000 is turned OFF again, it will move to the JME instruction without changing the status of output 1001.



JUMP / JUMP END

The following ladder diagram is similar to the Interlock (IL) example, except that it uses JUMP instruction. The ladder diagram and mnemonic code in Figure 4.19 represents JMP being used twice with one JME.



Address	Instruction	Data
0000	LD	0000
0001	JMP	#1
0002	LD	0001
0003	TIM	000
		#050
0004	LD	TIM000
0005	OUT	1001
0006	LD	0002
0007	JMP	#1
0008	LD	0003
0009	OUT	1002
0010	LD	0004
0011	LD	0005
0012	CNT	001
		#005
0013	JME	#1

Figure 4.19 – Application of JMP being used twice with one JME.

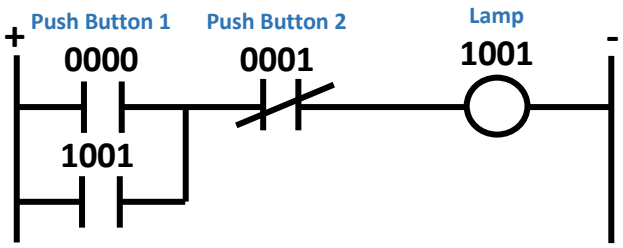
When input 0000 for 1st JMP is turned ON and 0002 for 2nd JMP is turned OFF, TIM000 will be activated for 5 seconds and output 1001 will be turned ON but CNT001 will not be changed and output 1002 will remain as current value.

When 0000 and 0002 for both JMP are ON, the program between the 1st JMP and the JME will execute normally.
Next, when both of JMP are turned OFF again. All the output relay, timer or counter remain unchanged.

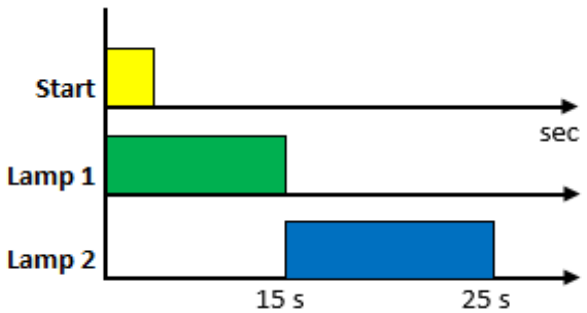
REFLECTION QUESTION

1. Explain the function of internal relay (IR) instruction using a ladder diagram and mnemonic code.

2. Based on the Figure 4.20, what happen to a circuit when push button **1** (**0000**) is turned ON and then, the push button **1** is turned OFF.



3. By using KEEP instruction set, draw the PLC ladder diagram based on the timing diagram shown in Figure 4.21.



4. Explain the operation of SET and RESET by referring to Figure 4.22.

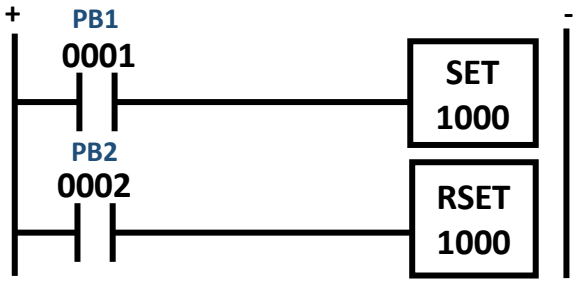


Figure 4.22

REFLECTION QUESTION

5. Design PLC ladder diagram by using KEEP and DIFD instruction to implement the timing diagram as shown in Figure 4.23.

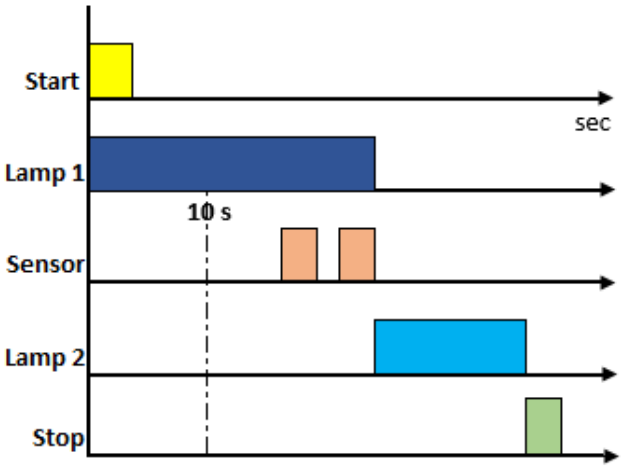


Figure 4.23

6. Explain the difference between DIFD and DIFU instruction.

7. Both IL and JMP instruction are to skip a section of instruction when they turn OFF. What are the differences between these two instructions when they are executed.

5.0 APPLICATION PLC PROGRAMMING

Application PLC Programming

1. Three Phase Motor Direct Control by using PLC
2. Three Phase Motor Forward-reverse Rotation Control By Using PLC
3. Pedestrian Crossing Traffic Light Control System
4. A Conveyor Dispatching System
5. Part Sorting For Assembled Material

THREE PHASE MOTOR DIRECT CONTROL BY USING PLC

3 phase motor can be controlled by PLC through a 3 phase contactor (K1) as in figure 5.1. The contactor contacts are closed when it is energized by applying 240V ac across its' coil . A normally open (N.O) push button switch is used to start the motor and a normally close (N.C) switch is used to stop the motor.

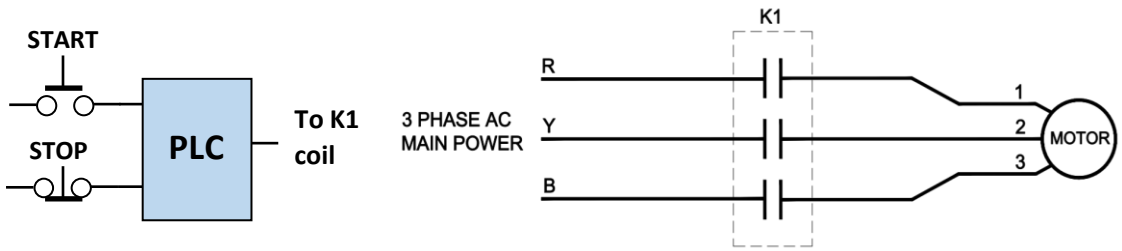


Figure 5.1 – 3 phase motor controlled using contactor

To write a control program for this process, we need to assign input and output terminal addresses to the input and output devices. Assuming that OMRON CPM2A is used, table 5.1 shows the assignment of PLC terminal address to device.

Table 5.1 – Input Output assignment

Device	Type	PLC Terminal Address
Start Switch (N.O)	Input	0000 or (0.00)
Stop Switch (N.C)	Input	0001 or (0.01)
Contactor Coil K1	Output	1000 or (10.00)

Since momentarily switch is used to start the motor, a holding or latching circuit is needed for the motor to be in the energized state even after the push button is released. To realise it, a internal relay bit or the output bit itself can be used to a accomplish the purpose as is shown in the ladder diagram 1 and 2.

Holding Circuit using Internal Relay

Referring to Figure 5.2, when start switch is pushed, bit 200.00 is activated through the start and the stop switches (stop contact is close because a N.C type is used for stop switch). The 2 associated contact of bit 200.00 will turn ON; one to hold the circuit in rung 1, another to drive the output as shown in rung 2.

When the stop button is pressed, its' contact 0001 opens to disconnect the flow of current to 200.00. The associated contacts of this bit thus open to release the latching action in rung 1 and deactivate the output 1000 in rung 2.

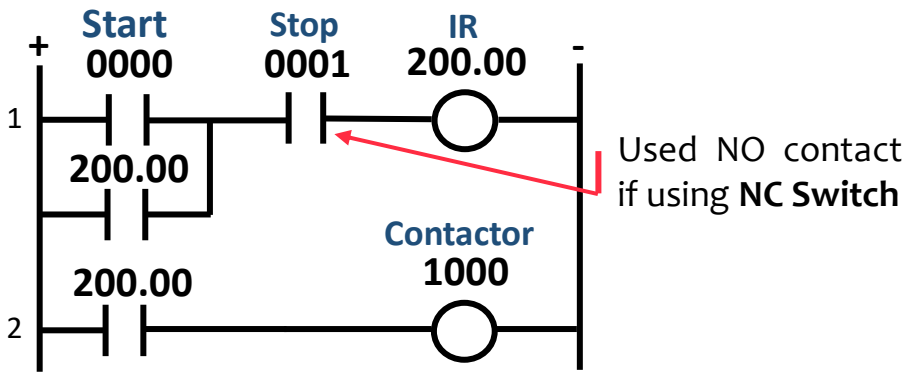


Figure 5.2 – Ladder Diagram 1

Holding Circuit using the Output Bit

Figure 5.3 shows the ladder diagram of another control by using the output bit associated contact itself as holding circuit.

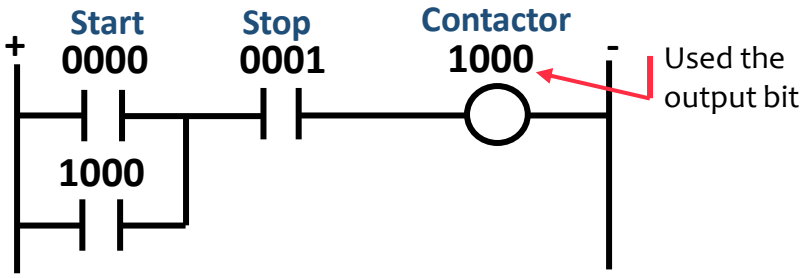


Figure 5.3 – Ladder Diagram 2

THREE PHASE MOTOR FORWARD-REVERSE ROTATION CONTROL BY USING PLC

Referring to figure 5.4, to have forward rotation, the R,Y and B lines are connected to motor terminals 1,2 and 3 respectively through contactor K1. To reverse the rotation, R,Y and B are connected to terminals 2, 1 and 3 respectively through contactor K2.

The control system for this process has 3 inputs and 2 output devices as shown in figure 5.4. 3 push button switches for stop, forward and reverse and 2 contactor coil K1 and K2.

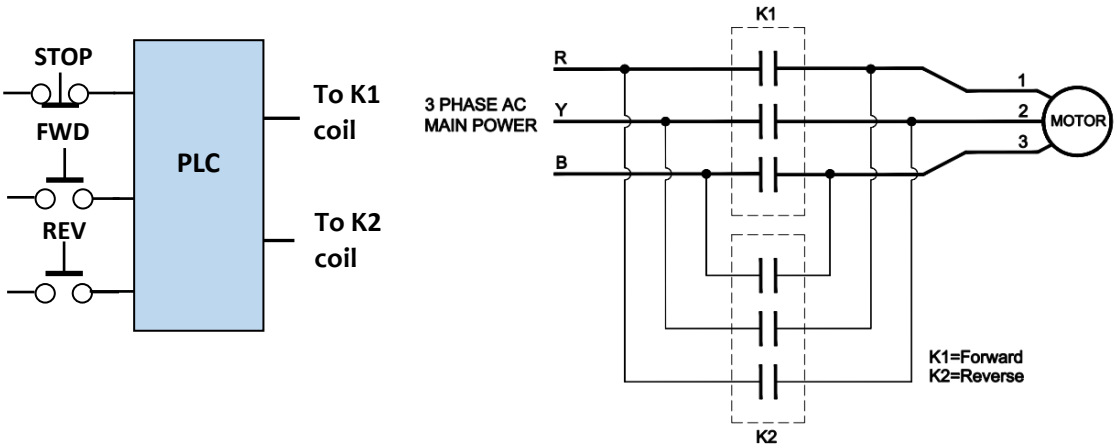


Figure 5.4 – Three Phase Motor Forward-reverse Rotation Control

Table 5.2 shows the assignment of PLC terminals addresses to input and output devices.

Table 5.2 – Input Output assignment

Device	Type	PLC Terminal Address
Stop Switch (N.C)	Input	0001
Forward (N.O)	Input	0002
Reverse (N.O)	Input	0003
Contactor Coil K1	Output	1000
Contactor Coil K2	Output	1001

Explanation based on the ladder diagram control program for the system.

Referring to the ladder diagram in Figure 5.5,

- Initially no contactor coil is energized, motor is in stop mode. When the forward button is pushed, K1 coil is energized through the stop switch (N.C), the FWD switch and the K2 associated normally close contact to run the motor in forward direction. Once K1 is energized, its associated contact in FWD rung will hold the circuit after FWD button is released.
- The associated contact of K1 in the REV rung is open because K1 is active is to prevent the energization of K2 coil if the REV button is accidentally pushed.
- The change the direction of motor rotation we need to stop the motor first, before we push the respective button.
- The operation of rotate the motor in reverse direction is similar to the forward action accept that the REV button is pushed, and K2 coil is energized instead of K1.

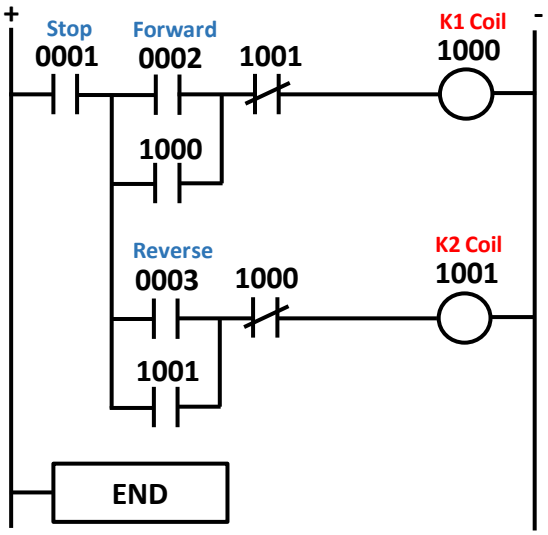


Figure 5.5 – Ladder Diagram

PEDESTRIAN CROSSING TRAFFIC LIGHT CONTROL SYSTEM

Figure 5.6 shows a pedestrian crossing system which operates as followed:

- When PLC is set to run mode, the Green light for car traffic and the Red light for pedestrian crossing are turned ON. It remains ON until the pedestrian crossing button is pressed and released.
- After a delay of 15 seconds, the Yellow light for car traffic turns ON and the Green light turns OFF. The pedestrian light remains Red.
- After another 5 seconds, car traffic light turns Red light. At the same time the pedestrian Green light turns ON for 5 second and **flickers 5 times (ON for 1 second and OFF for another 2 second**) and then Red light for car traffic remains ON.
- This condition remains for 2 seconds and then car traffic light turns Green light, and the pedestrian light remains Red light. The process will repeat when the button is pressed again.
- The button will only effective when the car traffic light is Green.

The summarized of the process operation as shown in Table 5.3.

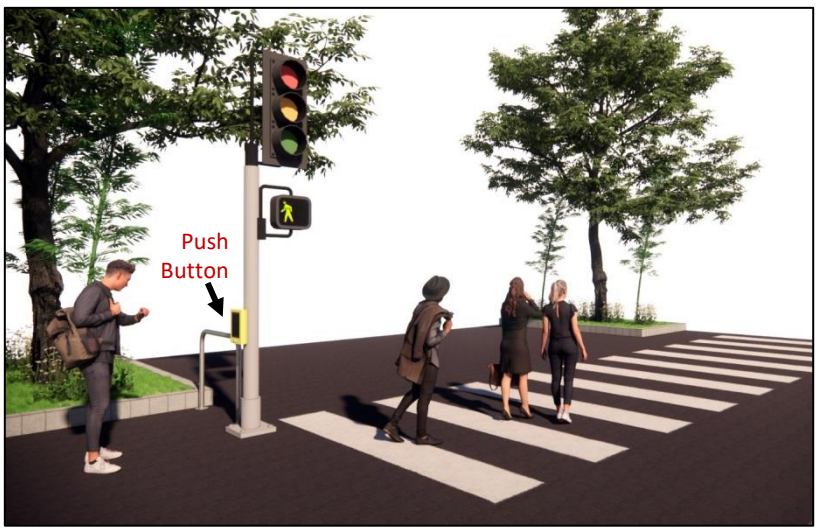


Figure 5.6 - Pedestrian Crossing System

Table 5.3 - Summarized the process operation

State	Car Traffic Light			Pedestrian Traffic Light		Time Delay (Sec)
	Green	Yellow	Red	Green	Red	
Initial State	ON	OFF	OFF	OFF	ON	
Button Pressed	ON	OFF	OFF	OFF	ON	15
After 15s	OFF	ON	OFF	OFF	ON	5
After 5s	OFF	OFF	ON	ON	OFF	5
After 5s	OFF	OFF	ON	FLICKER	OFF	5 times (off 2s on 1s)
After Flicker	OFF	OFF	ON	OFF	ON	2
After 2s	ON	OFF	OFF	OFF	ON	Wait for button press

Table 5.4 shows the input and output devices for the system and the assignment of PLC terminals to the devices.

Table 5.4 – Input Output assignment

Input Device	PLC Input Terminal
Push Button NO	0000

Output Devices	PLC Output Terminal
Red Car	1000
Yellow Car	1001
Green Car	1002
Red Pedestrian	1003
Green Pedestrian	1004

Figure 5.7 and Figure 5.8 shows the ladder diagram control program for the system.

In the initial condition, the Green light for the car traffic light is turned ON by TIM000 contact (NC contact) at rung 5 and the Red light for the pedestrian is turned ON by TM001 contact (NC contact) at rung 6.

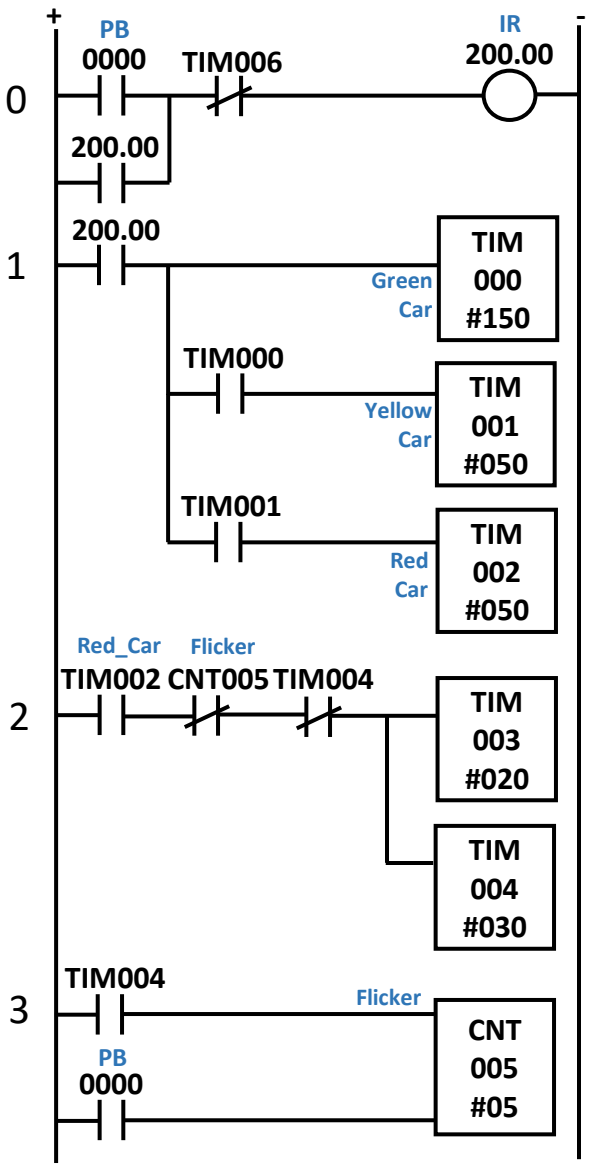


Figure 5.7 – Ladder Diagram (rung 0 – rung 3)

At rung 0, once the Pedestrian push button is pressed, TIM000 at rung 1 starts to operate. After 15 seconds, its associated contact TIM000 opens to turns OFF car Green traffic light at rung 5 and turn ON the Yellow light at rung 7.

In the same time, it also activates TIM001 at rung 1. 5 seconds after TIM001 activation, its' associated contact TIM001 turns OFF the pedestrian Red light at rung 6 and the Yellow car traffic light at rung 7, turns ON the car traffic Red light at rung 8 and the pedestrian green light at rung 9.

Continue to page 73

TIM001 also activated TIM002 at rung 1. TIM002 provides a time delay for 5 seconds before it activates the flicker circuit implemented by TIM003 and TIM004 at rung 2. The number of flickering is counted by CNT005 at rung 3.

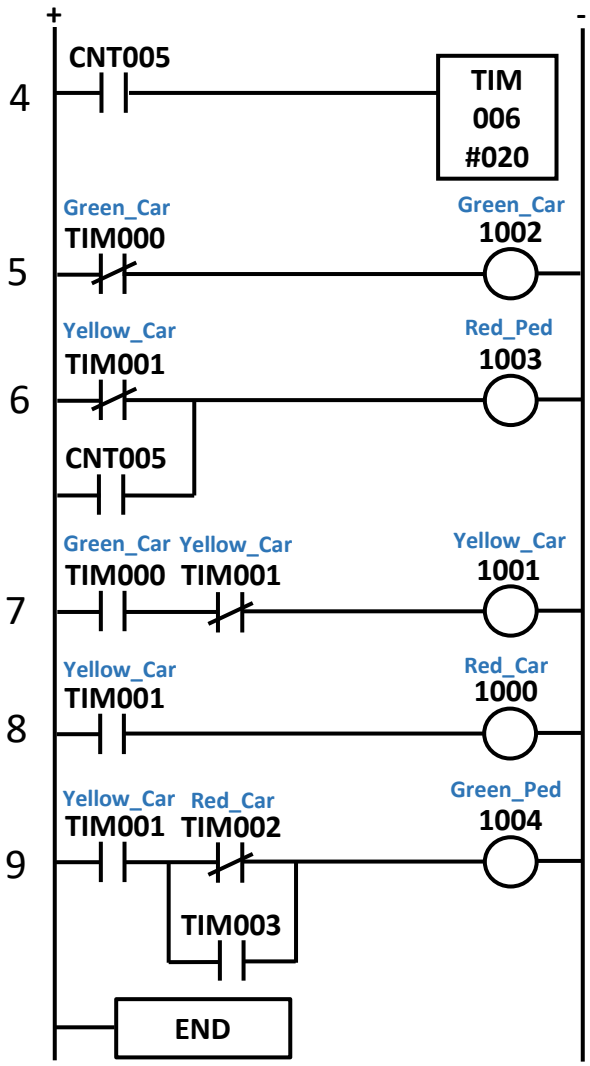


Figure 5.8 – Ladder Diagram (rung 4 – rung 9)

At each flicker, the associated contact TIM003 will turn OFF for 2 seconds and turn ON for 1 second. Thus making the pedestrian Green light to flicker at rung 9.

After 5 flickers, the associated contact of CNT005 is activated, cutting off the flickering circuit at rung 2, resetting TIM003 and TIM004 and causing the pedestrian Green light to turn OFF at rung 9. At the same instant, CNT005 contact also turns ON the pedestrian Red light at rung 6 as well as activates TIM006.

Since the associated contact of TIM006 is used to reset the process as shown in rung 0. After 2 seconds, the system is returned to its initial condition whereby the car Green traffic light and the pedestrian Red light are turned ON, waiting for the pedestrian to press the button again.

A CONVEYOR DISPATCHING SYSTEM

Figure 5.9 show a simple conveyor dispatching system of boxes. The system works as follow:

1. Press and release the 'start' Button (01) , motor (01) will rotate, and conveyor will move to carry the box towards the sensor. When the box on conveyor is detected by sensor (00) and motor stops immediately.
2. Once the Box has been taken away from the conveyor, motor will start to move after a delay of 5 seconds. Then process 1 and 2 are repeated.
3. When the tenth (10) Box is taken away from the conveyor, motor stops, and lamp (02) turns ON for 1 minutes and then turns OFF again.
4. If 'stop' button (02) is pressed , both motor and lamp will turn OFF immediately .

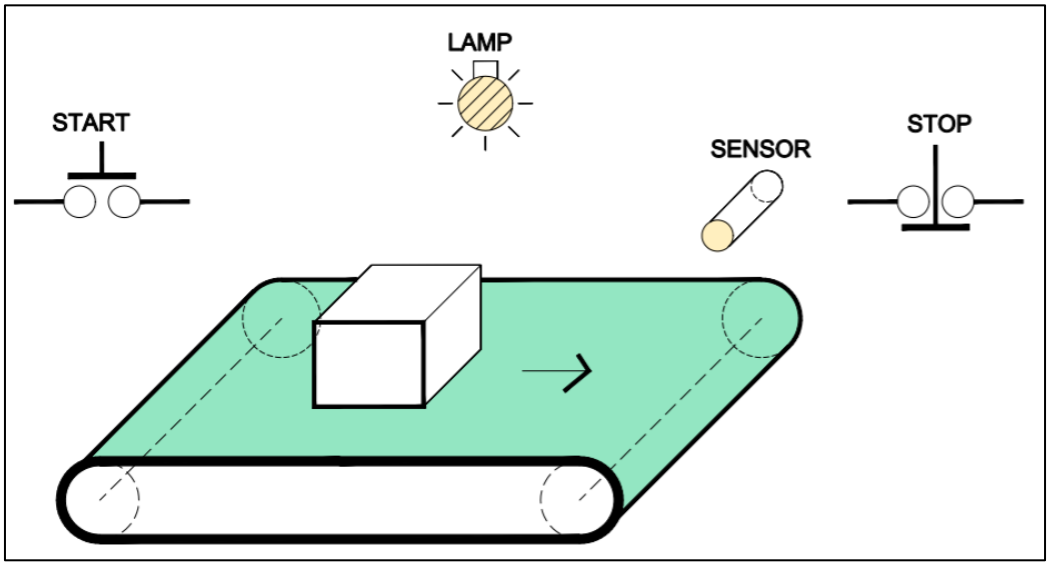


Figure 5.9 - Simple Conveyor Dispatching System Of Boxes

Assuming that OMRON CPM2A is used, table 5.5 shows the assignment of PLC terminal address to devices.

Table 5.5 – Input Output assignment

	Input device	PLC input terminal	Output devices	PLC output terminal
1	Start Button N/O	0001	Motor	1001
2	Stop Button N/C	0002	Lamp	1002
3	Sensor	0000		

The solution for the control process is shown as ladder diagram in Figure 5.10.

- Since a momentarily switch is used to start the process, we need a latching circuit to hold on the start process when the push button of the start switch is released.
- In this case a temporary bit (200.00) is used to fulfill this purpose as shown in rung 0.
- The closure of the associated contact of bit 200.00 causes the conveyor motor which is driving by PLC output 10.01 to start running as shown in rung 1.
- A holding circuit for the sensor is also needed to prevent the conveyor from moving once the sensor is not detecting the box when it is taken away from the conveyor. IR 200.01 is used for this purpose as in rung 2.
- The associated contact of IR 200.01 bit is used to stop the conveyor during this time as shown in rung 1.

- Since the sensor input is latched, we need a way to find out when does the box is removed from the conveyor. A DIFD is used to generate a 1 cycle pulse when the sensor contact transit from high to low while the box is removed. This is shown in rung 3.
- The generated pulse causes bit 200.02 associated contact to close and open in a cycle time. This momentarily close is latched by bit 200.03 to energized TIM000 as in rung 4.
- A time delay of 5 seconds is implemented by TIM000, and its' associated contact (TIM000) is used to reset itself in rung 4 and to cut off the latching of sensor in rung 2.

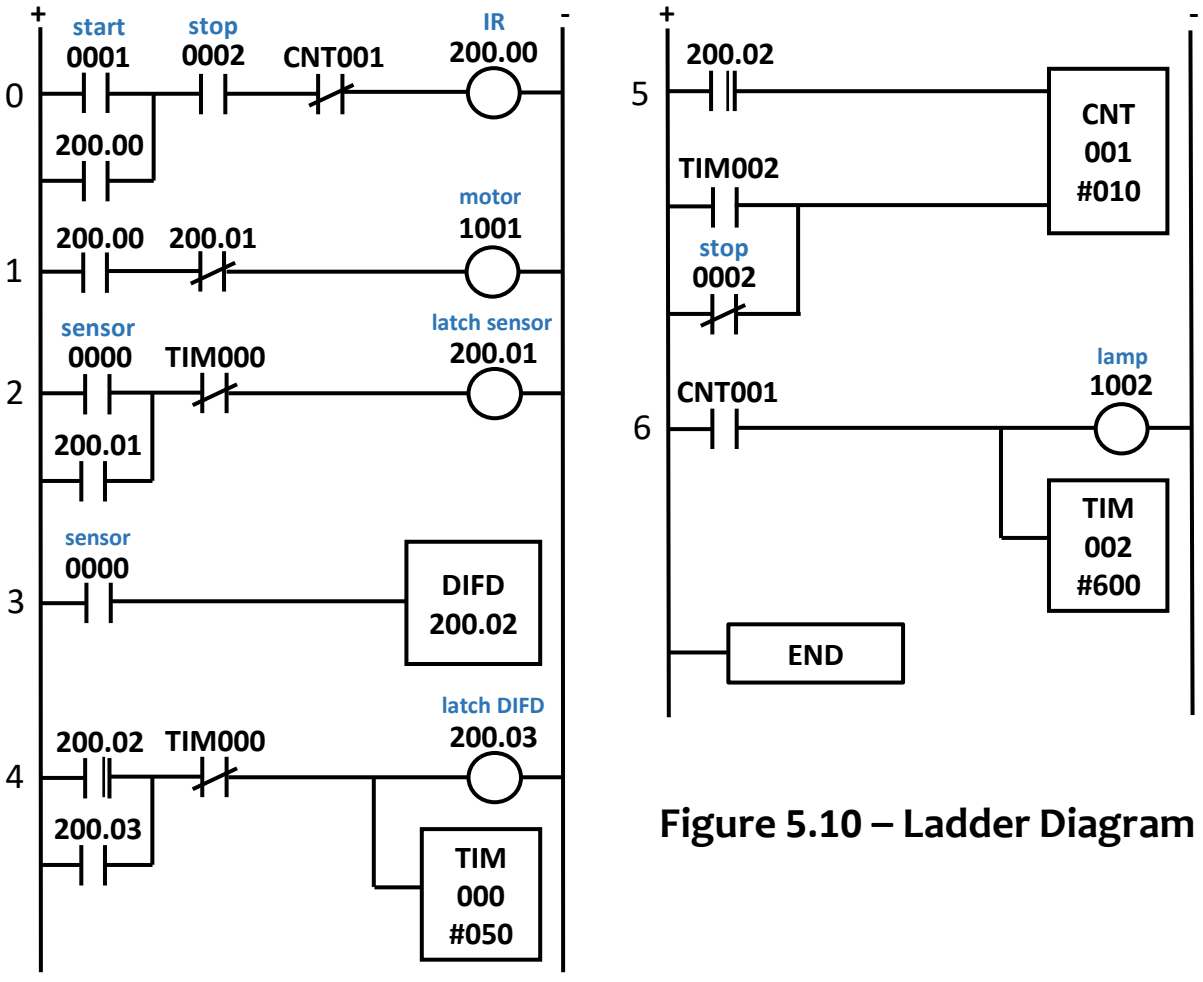


Figure 5.10 – Ladder Diagram

- The unlatching of sensor contact bit 200.01 in turn causes its associated contact in rung 1 to close back and moving the conveyor once again.
- The pulses generated by bit 200.02 when a box is removed from the conveyor is used as input to CNT001 as in rung 5. The counter value is set at a value of 10 for its associated contact CNT001 to operate once the tenth box is removed from the conveyor.
- This contact will set output bit 1002 to turn ON the lamp and at the same time energized TIM002 as shown in rung 6. The same associated contact is also used to reset bit 200.00 in rung 0 and in turn reset output bit 1001 in rung 1 to stop the conveyor.
- After a time, delay of 1 minute, TIM002 associated contact TIM002 is used to reset the counter so that contact CNT001 becomes open to turn off the lamp and TIM002 as well.
- The stop contact 0002 is used to stop the process at any time when the stop button is pressed. At the same time, it is also used to reset the counter.

https://youtu.be/G4l-frzBzCU

You also can watch this video to understand how to write the programming for this application.

Just click this link or scan the QR code.

PART SORTING FOR ASSEMBLED MATERIAL

Figure 5.11 and figure 5.12 shows a system that can be used to sort out assembled part from unassembled metal or plastic part.

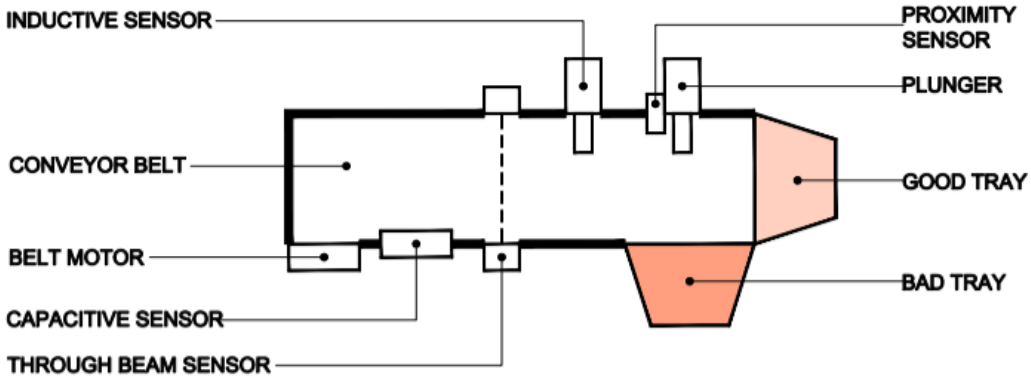


Figure 5.11 - System Top View

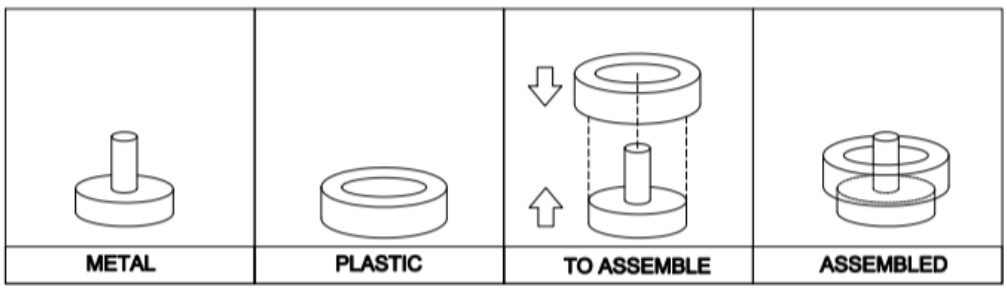


Figure 5.12 - Parts

- When start switch is pressed and released, the conveyor belt (motor) will move.
- A part is placed on the belt so that it will go through the sensors along the way from left to right.
- At the end of the journey, it will either ejected by the plunger into the Bad tray if the part is not assembled or otherwise it will go into the good tray.

The input devices (sensors)

Capacitive sensor : Capacitive sensor can be used to detect nonmetallic or metallic part. As long as there is part about a few mm in front of the sensor head, the sensor will turn ON. Since it is mounted a few cm above the belt, it can be used to sense the assembled plastic part.

Inductive sensor : Can only sense the metal piece. It will turn on when there is a metal piece a few mm in front of it.

Through beam sensor : It will turn on when there is object blocking the light beam.

Proximity sensor : It will turn on when there is object in front of it.

The output devices

Motor : This motor is used to drive the conveyor belt in order to carry component from the left side to the right side of the conveyor system.

Plunger : The plunger is activated by the solenoid. Once the solenoid is energized, it will extend to eject what ever object in front of it.

Table 5.6 shows the assignment of PLC terminal addresses to input and output devices of the system.

Table 5.6 – Input Output assignment

Input Devices	PLC Output Terminal
Start Switch	0000
Stop (N.O)	0001
Inductive Sensor (NPN)	0002
Capacitive Sensor (NPN)	0003
Through Beam sensor (NPN)	0004
Proximity Sensor (NPN)	0005

Output Device	PLC Input Terminal
Motor (DC 24V)	1004
Plunger (Solenoid)	1003

Explanation based on the ladder diagram control program for the system.

- 3 sensors are used to detect the part move along the belt. If the part is not assembled, one or two of the sensor will not be activated.

- The capacitive sensor will detect the plastic assembled on top of the metal piece.

- The inductive sensor will detect the lower part of a metal piece. The through beam sensor will detect any part that interrupt its' light beam. If the part is not assembled, only the capacitive sensor would not be activated.

- If it is a plastic part, only through beam sensor will be activated and if it is metal part, both inductive and through beam sensors will be activated.

The summarized of the detection outcomes as shown in Table 5.7.

Table 5.7 - Summarized the detection outcomes.

Condition	Inductive (200.01)	Capacitive (200.02)	Through beam (200.03)	Conclusion
1	ON	ON	ON	Assembled part
2	OFF	ON	ON	2 plastic stack up
3	OFF	OFF	ON	Plastic part
4	ON	OFF	ON	Metal part

The function KEEP used in ladder diagram is to start and stop the system process. When the start button is pressed the system is latched until the stop button is pressed.

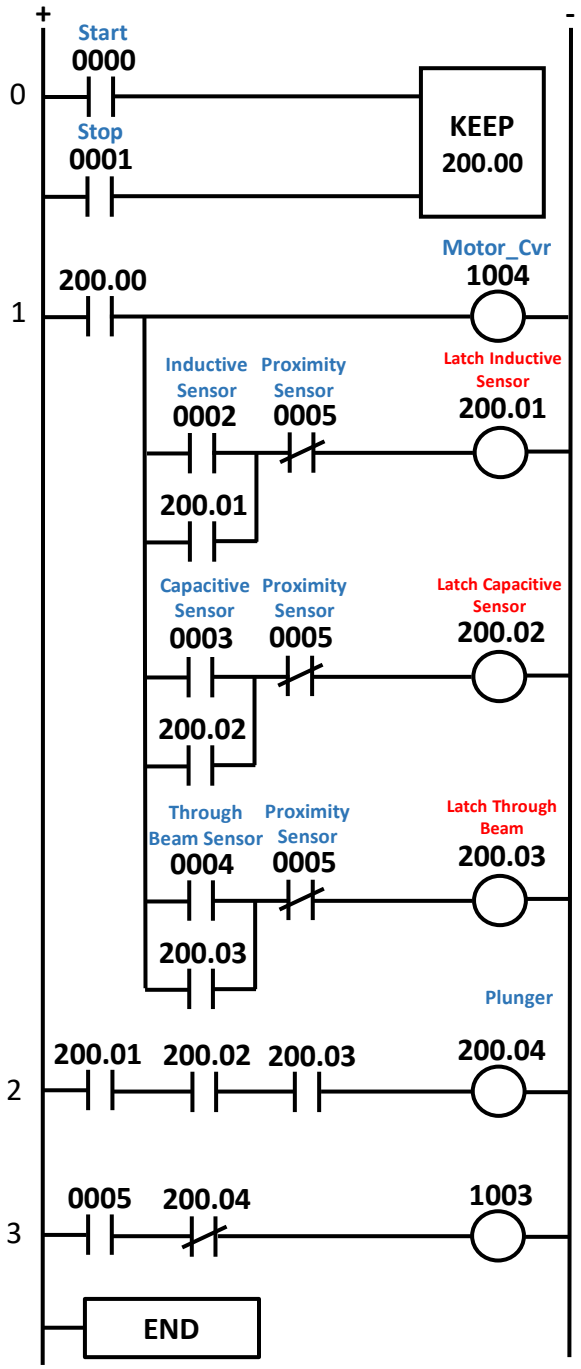


Figure 5.13 – Ladder Diagram

The condition of the sensor is used to latch a temporary bit as shown in rung 1. Inductive sensor will latch bit 200.01 if it sensed a metal piece. Capacitive sensor will latch bit 200.02 if it sensed a assembled plastic and through beam sensor will latch 200.03 if there is an interruption on its' light beam path.

Rung 3 shows how the plunger work. If a part is detected by proximity sensor at the end of the conveyor (contact 0005 will close), the activation of the plunger will depend on the condition at rung 2. All conditions will activate the plunger to eject the part into the bad tray except condition 1 (refer table 5.7), the assembled part will go into good tray.

REFLECTION QUESTION

Figure 5.14 shows a filling and draining system. Based on the sequence operation below, design PLC ladder diagram by using KEEP and DIFD instruction.

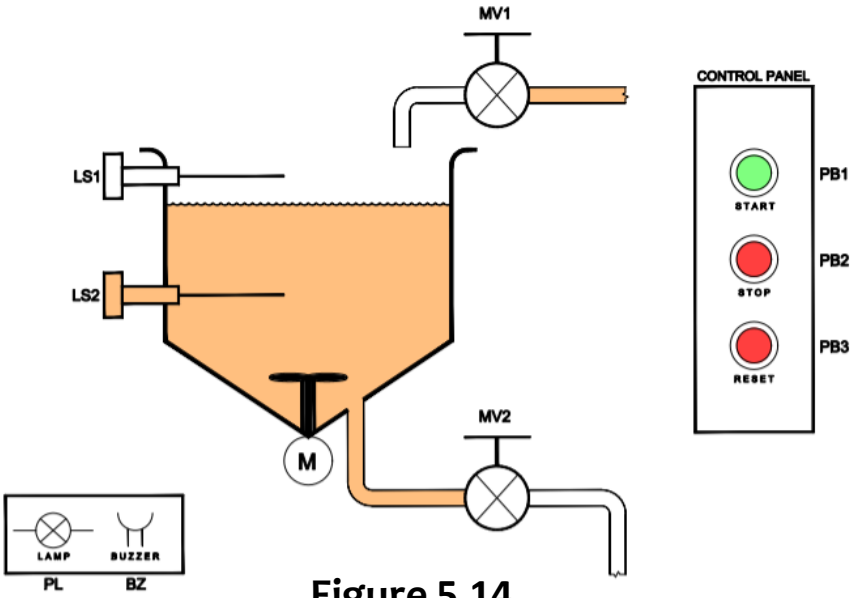


Figure 5.14

i. If the start button (PB1) is pressed, motor valve 1 (MV1) is opens and the water begins to fill the tank. At the same time, the stirring motor (M) starts operation.

ii. When the water level passed to sensor LS2 and reaches sensor LS1, MV1 closes and the stirring motor stop. Next, motor valve 2 (MV2) is opens and starts draining the water.

iii. When the water level drops below LS2, MV2 closes. After the cycle of operation has repeated four times, the indicator lamp is illuminated, and the buzzer operates for 5 seconds. The operation can be restarted after the stop button (PB2) is pressed.

REFERENCE

[1] Petruzella, F. D. (2017). Programmable Logic Controllers. New York. McGraw-Hill Companie

[2] W. Bolton (2015). Programmable Logic Controllers. Elsevier Ltd.

[3] CPM2A Programmable Controllers Operation Manual (1999). Omron

[4] A Beginners Guide to PLC. (2015). Singapore: Omron

The Programming Logic Controllers (PLC) Programming and Applications eBook is aimed to equip students with the knowledge and skills related to basic PLC programming and it's application in industry.

POLITEKNIK MALAYSIA
UNGKU OMAR

e ISBN 978-967-2421-35-1

9 7 8 9 6 7 2 4 2 1 3 5 1