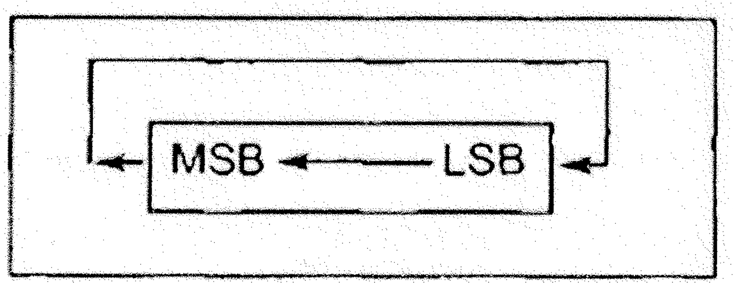
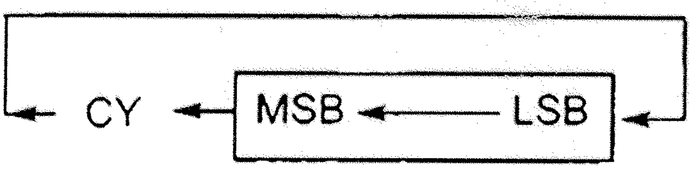
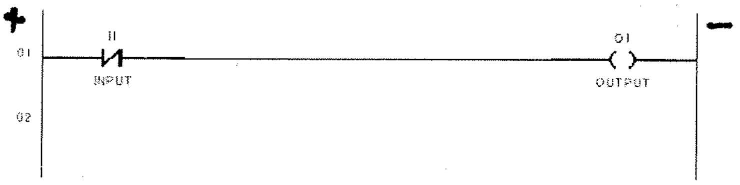
SCORING INDICATORS (VERSION B)

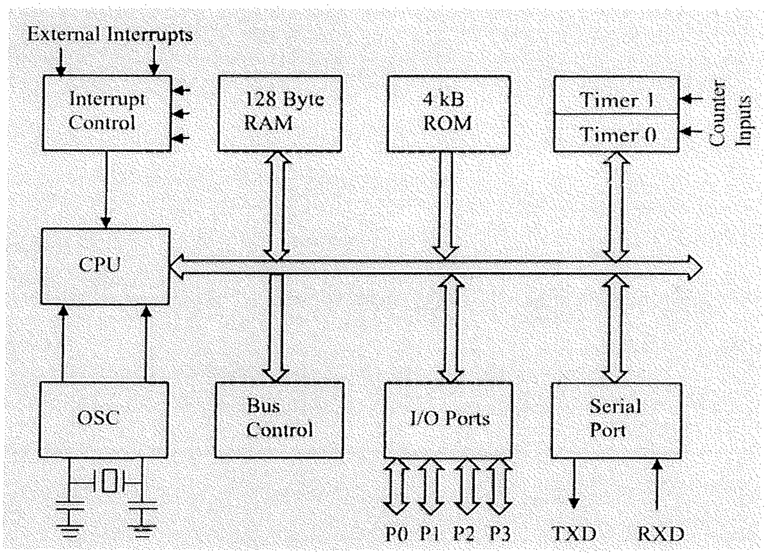
COURSE NAME : MICROCONTROLLER &PLC

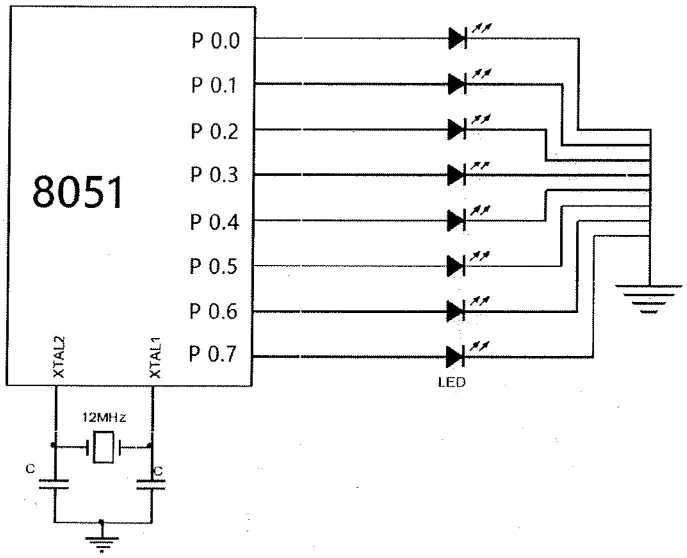
COURSE CODE : 6031C **QID : 2102240047**

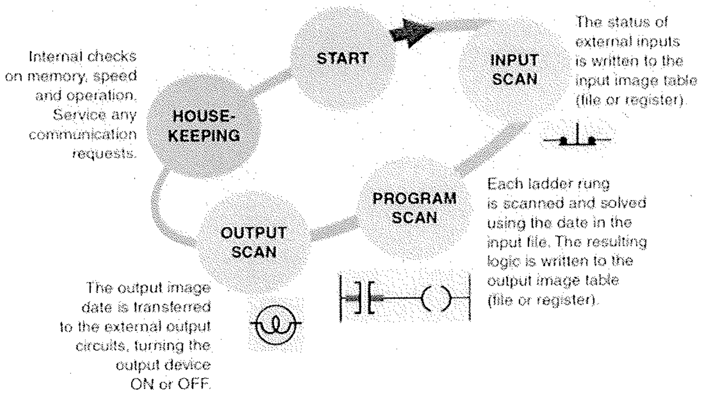
Q No	Scoring Indicators	Split score	Sub Total	Total Score
	PART A			9
I. 1	4K Bytes	1	1	
I. 2	Data pointer Program counter Timer 0 Timer 1 Any two	0.5*2	1	
I. 3	Register indirect addressing mode	1	1	
I. 4	RL is simple rotation of the bits of A to left, and RLC is rotation of bits ofA to left through the carry. **RL** 	0.5*2	1	

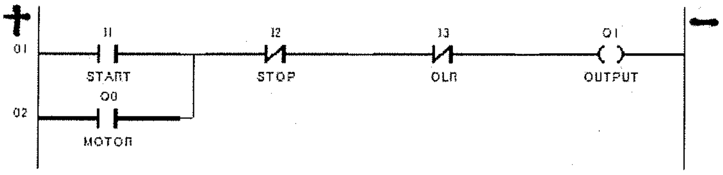
	RLC 			
I. 5	PLC stands for Programmable Logic Controllers.It is a digitally operating electronic apparatus which uses a programming memory for the internal storage of instructions for implementing specific functions to control various types of machines or process in automated system . Various definitions can be accepted			
I. 6	Flexibility,Reliability,Real Time Operation,Programmable, can be integrated with wide range of input & output devices,cost effective Any two	0.5*2	1	
I.7	The Transportation System likes Conveyor Belt System. It is used in the Power Generation, Transmission, and Distribution System. Smart Traffic Control Signal System Fire Detection and Alarm System. Automatic Bottle or Liquid Filling System. Any two	0.5*2	1	
I. 8	Rails	1	1	
I. 9		1	1	

Q No	Scoring Indicators	Split score	Sub Total	Total Score
	PART B			30
II. 1	PSW Program Status Word Register 7 6 5 4 3 2 1 0 CY AC FO RS1 RSO OV – P The program status word (PSW) register is an 8-bit register. It is also referred to as the flag register. - CY: Carry flag. This flag is set whenever there is a carry out from the D7 bit after an 8 bit addition or subtraction. - AC: Auxiliary carry flag If there is a carry from D3 and D4 during an ADD or SUB operation, this bit is set; - F0: Available to the user for general purposes. - RS0, RS1: Register bank selects bits These two bits are used to select one of the four register banks from internal RAM	3	3	
II. 2	Interrupts are the events that temporarily suspend the main program, pass the control to the external sources and execute their task. It then passes the control to the main program where it had left off. 5 Interrupts in 8051 (with order of normal priority) 1. External hardware interrupts 2. $\overline{INTO}$ 3. Timer 0 overflow interrupt T0 4. External hardware interrupts $\overline{INT1}$ 5. Timer 1 overflow interrupt T1 6. Serial interrupt (TI & RI) RESET- It is the ultimate interrupt in 8051 and is non maskable . Whenever a high level is applied to RST pin , the 8051 enters a reset condition (Def 1+Listing 2)	1+2	3	

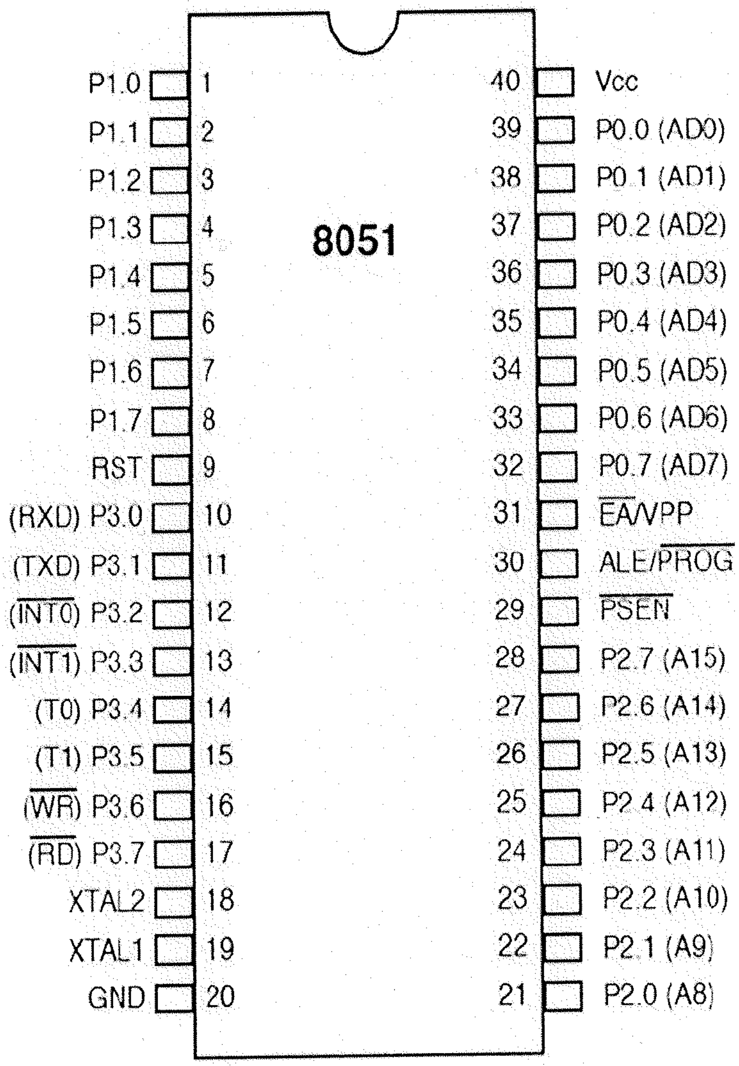
II. 3	 External Interrupts Interrupt Control 128 Byte RAM 4 kB ROM Timer 1 Timer 0 Counter Inputs CPU OSC Bus Control I/O Ports Serial Port P0 P1 P2 P3 TXD RXD	3	3	
II. 4	Consumer Appliances (TV Tuners, Remote controls, Computers, Sewing Machines, etc.) Home Applications (TVs, VCR, Video Games, Camcorder, Music Instruments, Home Security Systems, Garage Door Openers, etc.) Communication Systems (Mobile Phones, Intercoms, Answering Machines, Paging Devices, etc.) Office (Fax Machines, Printers, Copiers, Laser Printers, etc.) Automobiles (Air Bags, ABS, Engine Control, Transmission Control, Temperature Control, Keyless Entry, etc) Aeronautical and Space Medical Equipment Defense Systems Robotics Industrial Process and Flow Control Radio and Networking Equipment Remote Sensing (Any 6)	6*0.5	3	

II. 5	MOV A, # 53 H ;[A]=53 H SWAP A ;[A]=35H DEC A ;[A]=34H			
II. 6	 8051 P 0.0 P 0.1 P 0.2 P 0.3 P 0.4 P 0.5 P 0.6 P 0.7 XTAL2 XTAL1 LED 12MHz C C 1. Anodes of the LEDs are Connected to the port pins and cathodes are connected to common ground connection. 2. To turn on particular LED we will need to make value of that pin "High" i.e "1". 3.After making a particular pin high or low a small delay is executed to make that LED light visible.	3	3	
II.7	Input devices DIP switch push button reed switch, limit switch, light sensors proximity switch Output devices Contractors	0.5*6	3	

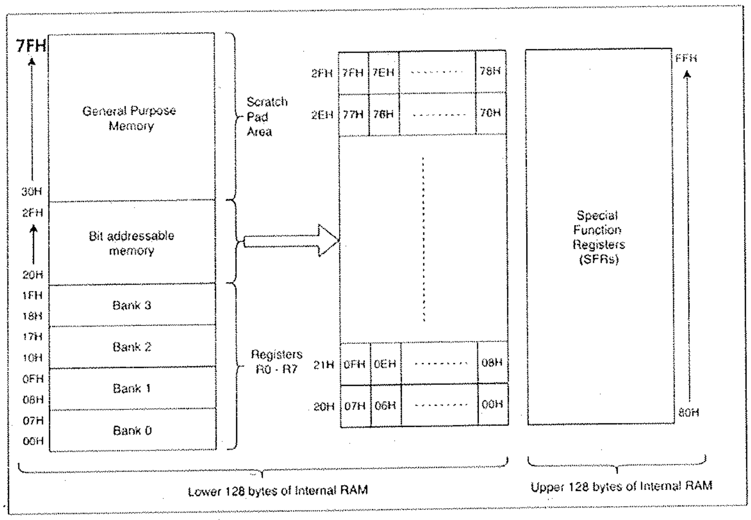
	pilot lamps Relays Alarms solenoid (Any 3 input device&3 output device)			
II. 8	*System (task) requirements. * Application requirements. * input/output capacity * type of inputs/outputs * size of memory * speed of CPU * Electrical requirements. * Speed of operation. * Communication requirements. * programming requirement&Software. * Operator interface. * Physical environments. (any 6)	0.5*6	3	
II. 9	PLC performs its operation operation based on a scan cycle  START INPUT SCAN The status of external inputs is written to the input image table (file or register). PROGRAM SCAN Each ladder rung is scanned and solved using the date in the input file. The resulting logic is written to the output image table (file or register). OUTPUT SCAN The output image date is transferred to the external output circuits, turning the output device ON or OFF. HOUSE-KEEPING Internal checks on memory, speed and operation. Service any communication requests.	2+1	3	

	The PLC program is executed as part of a repetitive process referred to as a scan. A typical PLC scan starts with the CPU reading the status of inputs. Next, the application program is executed. Next, the CPU performs internal diagnostic and communication tasks. Finally, the status of all outputs is updated. This process is repeated continuously as long as the PLC is in the run mode.			
II. 10	 + 01 I1 START O0 02 MOTOR I2 STOP I3 OLR O1 OUTPUT − Inputs: START- To start the motor- NO STOP - To stop the motor-NC OLR- Overload Relay Output: MOTOR- Motor	2+1	3	

	Sequence of Events : 1. When the Start push button is pressed, Motor has to start. 2. If the Start is released and the Stop pushbutton is not pressed, Motor should remain on. 3. When the Stop push button is pressed, Motor has to stop. 4. If stop push button is released and start is not pressed(released) motor should remain off. 5.When an overload occurs the motor stops			

Q No	Scoring Indicators	Split score	Sub Total	Total Score
	PART C			84
III	 8051 P1.0 1 P1.1 2 P1.2 3 P1.3 4 P1.4 5 P1.5 6 P1.6 7 P1.7 8 RST 9 (RXD) P3.0 10 (TXD) P3.1 11 ($\overline{\text{INT0}}$) P3.2 12 ($\overline{\text{INT1}}$) P3.3 13 (T0) P3.4 14 (T1) P3.5 15 ($\overline{\text{WR}}$) P3.6 16 ($\overline{\text{RD}}$) P3.7 17 XTAL2 18 XTAL1 19 GND 20 40 Vcc 39 P0.0 (AD0) 38 P0.1 (AD1) 37 P0.2 (AD2) 36 P0.3 (AD3) 35 P0.4 (AD4) 34 P0.5 (AD5) 33 P0.6 (AD6) 32 P0.7 (AD7) 31 $\overline{\text{EA}}$/VPP 30 ALE/$\overline{\text{PROG}}$ 29 $\overline{\text{PSEN}}$ 28 P2.7 (A15) 27 P2.6 (A14) 26 P2.5 (A13) 25 P2.4 (A12) 24 P2.3 (A11) 23 P2.2 (A10) 22 P2.1 (A9) 21 P2.0 (A8) **Pins 1 to 8** – These pins are known as Port 1. This port doesn't serve any other functions. It is internally pulled up, bi-directional I/O port.	5+2	7	

	Pin 9 – It is a RESET pin, which is used to reset the microcontroller to its initial values. **Pins 10 to 17** – These pins are known as Port 3. This port serves some functions like interrupts, timer input, control signals, serial communication signals RxD and TxD, etc. **Pins 18 & 19** – These pins are used for interfacing an external crystal to get the system clock. **Pin 20** – This pin provides ground to the circuit. **Pins 21 to 28** – These pins are known as Port 2. It serves as I/O port. Higher order address bus signals are also multiplexed using this port. **Pin 29** – This is PSEN pin which stands for Program Store Enable. It is used to read a signal from the external program memory. **Pin 30** – This is ALE pin which stands for Address Latch Enable. It is used to demultiplex the address-data signal of port. During Flash Programming, this pin acts as program pulse input (PROG). **Pin 31** – This is EA pin which stands for External Access input. It is used to enable/disable the external memory interfacing. i.e. allows external Program Memory. Code from external program memory can be fetched only if this pin is LOW. For normal operations, this pins is pulled HIGH. **Pins 32 to 39** – These pins are known as Port 0. It serves as I/O port. Lower order address and data bus signals are multiplexed using this port. **Pin 40** – This pin is used to provide power supply to the circuit.			

<table>
<tr><td>IV</td><td>

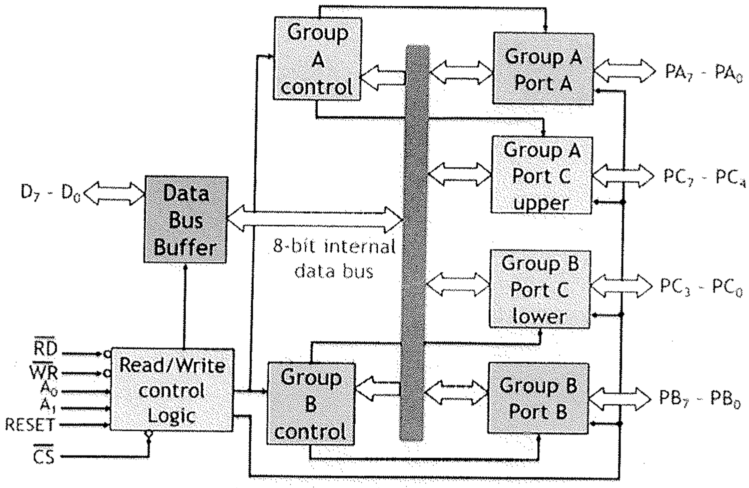
Register Banks (Bank 0,1,2,3) from 00H to 1FH – 32 bytes
Bit Addressable Area from 20H to 2FH – 16 bytes
General Purpose Register (Scratch Pad Area) from 30H to 7FH – 80 bytes
Upper 128 bytes (80H – FFH) for the Special Function Register (SFRs) which includes I/O ports (P0, P1, P2, P3), Accumulator (A), Timers(THx, TLx, TMOD, TCON, PCON), Interrupts(IE, IP), Serial Communication controls(SBUF, SCON), Program Status Word (PSW).
fig(4)explanation(3)</td><td>4+3</td><td>7</td><td></td></tr>
<tr><td>V</td><td>ALGORITHM
1.Start
2.Input lower and higher byte of 1st number to R0& R1
3.Input lower and higher byte of 2nd number to R2 &R3
4.Add the content of R2 and R0 register
5.Add the content of R1 and R3 register WITH CARRY
6.Store the result Stop the program</td><td>3+4</td><td>7</td><td></td></tr>
</table>

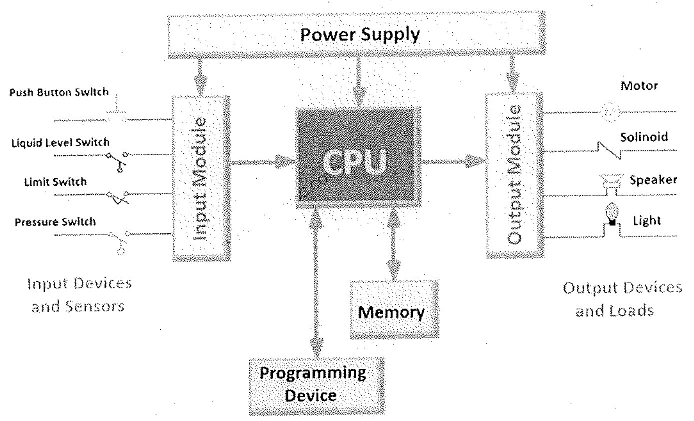
PROGRAM:

```
 ORG:5000
 MOV DPTR, #8000  ;   [ DPTR]=8000
 MOVX A,@DPTR     ;   [A]=[8000]
 MOV R0,A         ;   [R0]<----[A]
 INC DPTR         ;   [DPTR]=8001
 MOVX A,@DPTR     ;   [A]=[8001]
 MOV R1,A         ;   [R1]<-----[A]
 INC DPTR         ;   [DPTR]=8002
 MOVX A,@DPTR     ;   [A]=[8002]
 MOV R2,A         ;   [R2]<-----[A]
 INC DPTR         ;   [DPTR]=8003
 MOVX A, @DPTR    ;   [A]=[8003]
 MOV R3,A         ;   [R3]<-----[A]
 MOV A,R0         ;   [A]<-----[R0]
 ADD A,R2         ;   [A]= [A]+[R2]
 INC DPTR         ;   [DPTR]=8004
 MOVX @DPTR,A     ;   [8004]<-----[A]
 MOV A,R1         ;   [A]<------[R1]
 ADDC A,R3        ;   [A]<-------[A]+[R3]+C
 INC DPTR         ;   [DPTR]=8005
 MOVX @DPTR,A     ;   [8005]<-----[A]
HERE:SJMP HERE
```

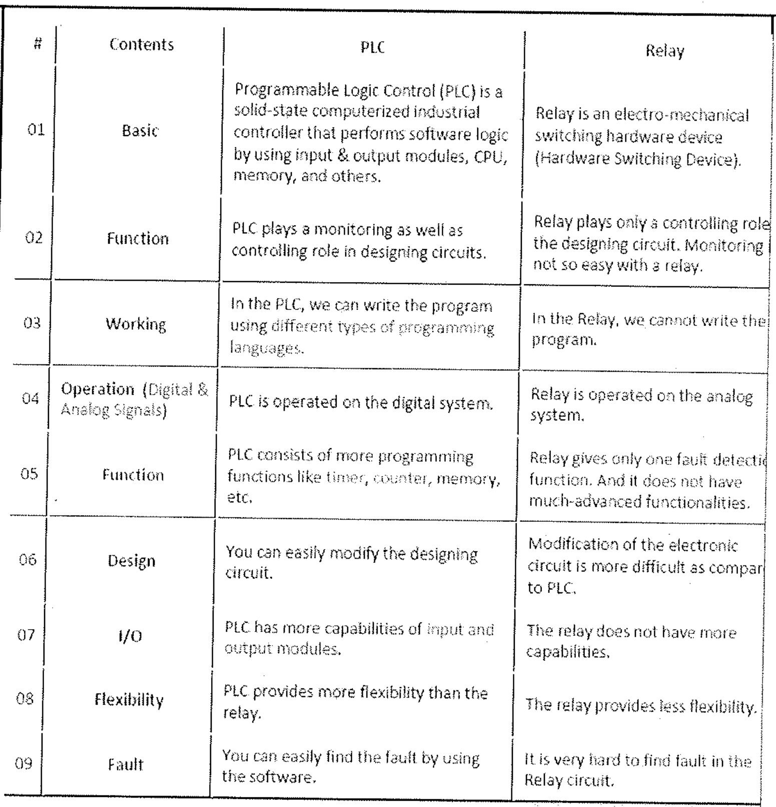
algorithm(3)+program(4)

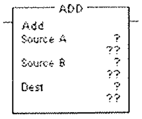
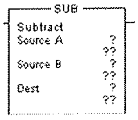
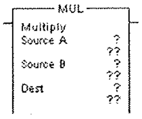
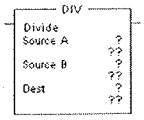
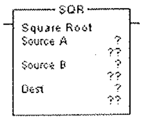
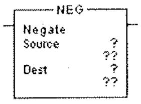
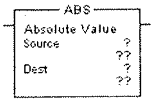
<table>
<tr>
<td>VI</td>
<td>• Immediate Addressing Mode
In this Addressing Mode, the data is provided in the instruction itself. The data is provided immediately after the opcode
• Register Addressing Mode
In this mode the source or destination data should be present in a register (R0 to R7).
• Direct Addressing Mode
In this Mode, the source or destination address is specified by using 8-bit data in the instruction.
• Register Indirect Addressing Mode
In this mode, the source or destination address is given in the register. By using register indirect addressing mode, the internal or external addresses can be accessed.
• Indexed Addressing Mode
In this mode, the source memory can only be accessed from program memory only. The destination operand is always the register A.
• Implied Addressing Mode
In this mode, there will be a single operand. These types of instruction can work on specific registers only.
<table>
<tr><td>ADD A,@RO</td><td>register indirect addressing moc</td></tr>
<tr><td>SUBB A,#45H</td><td>immediate addressing mode</td></tr>
<tr><td>ANL A,R2</td><td>register addressing mode</td></tr>
<tr><td>MOVC A,@A+DPTR</td><td>Register indexed addressing mode</td></tr>
<tr><td>RLA</td><td>implied addressing mode</td></tr>
<tr><td>MOV R2,45H</td><td>direct addressing mode</td></tr>
</table>
</td>
<td>4+3</td>
<td>7</td>
<td></td>
</tr>
</table>

VII	 Group A control Group A Port A PA₇ - PA₀ Group A Port C upper PC₇ - PC₄ D₇ - D₀ Data Bus Buffer 8-bit internal data bus Group B Port C lower PC₃ - PC₀ RD WR A₀ A₁ RESET CS Read/Write control Logic Group B control Group B Port B PB₇ - PB₀ Architecture of 8255 PPI Electronics Desk 8255A has three ports, i.e., PORT A, PORT B, and PORT C. •**Port A** contains one 8-bit output latch/buffer and one 8-bit input buffer. •**Port B** is similar to PORT A. •**Port C** can be split into two parts, i.e. PORT C lower (PC0-PC3) and PORT C upper (PC7-PC4) by the control word. •These three ports are further divided into two groups, i.e. Group A includes PORT A and upper PORT C. •Group B includes PORT B and lower PORT C. •***Data bus buffer***: It is a bidirectional data bus used to connect the internal bus of 8255 with the system bus so as to establish proper interfacing between the two. •**Read/ Write control logic**: This unit manages the internal operations of the system. fig(4)explanation(3)	4+3	7	
VIII	Logical Instructions, which perform logical operations like AND, OR, XOR, NOT, Rotate, Clear and Swap. Logical Instruction are performed on Bytes of data on a bit-by-bit basis. **AND OPERATION** ANL A, Rn ;[A]<-[A] AND [Rn]	7*1	7	

	ANL A, Address ;[A]<-[A] AND [Data at Address] ANL A, @Rn ;[A]<-[A] AND [Data at Address in Rn] ANL A, #data ;[A]<-[A] AND [Data] **OR OPERATION** ORL A, Rn ;[A]<-[A] OR [Rn] ORL A, Address ;[A]<-[A] OR [Data at Address] ORL A, @Rn ;[A]<-[A] OR [Data at Address in Rn] ORL A, #data ;[A]<-[A] OR [Data] **XOR OPERATION** XRL A, Rn ;[A]<-[A] XOR [Rn] XRL A, Address ;[A]<-[A] XOR [Data at Address] XRL A, @Rn ;[A]<-[A] XOR [Data at Address in Rn] XRL A, #data ;[A]<-[A] XOR [Data] CLR A,CPL A RL A,RR A,RRC A,RLC A,SWAP A			
IX	 Power Supply Push Button Switch Liquid Level Switch Limit Switch Pressure Switch Input Module CPU Output Module Motor Solinoid Speaker Light Input Devices and Sensors Memory Programming Device Output Devices and Loads Programmable Logic Controller or PLC Block Diagram	4+3	7	

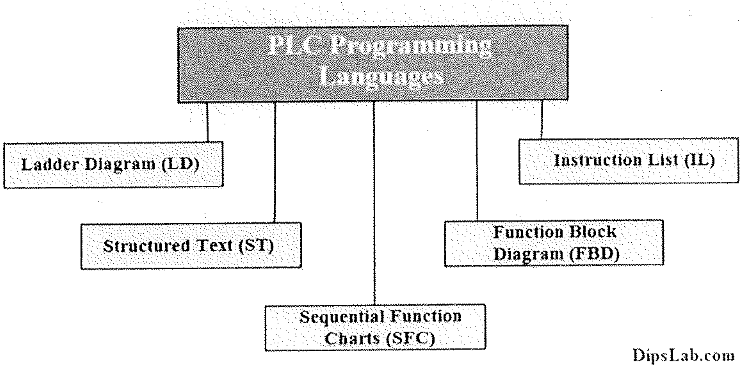
	Input/ Output Section: The input section or input module consists of devices like sensors, switches, and many other real-world input sources. The input from the sources is connected to the PLC through the input connector rails. The output section or output module can be a motor or a solenoid or a lamp or a heater, whose functioning is controlled by varying the input signals. • CPU or Central Processing Unit: It is the brain of the PLC. It can be a microprocessor.It carries out all the processing related to the input signals in order to control the output signals based on the control program. • Programming Device: It is the platform where the program or the control logic is written. It can be a handheld device or a laptop or a computer itself. • Power Supply: It generally works on a power supply of about 24 V, used to power input and output devices. • Memory: The memory is divided into two parts- The data memory and the program memory. The program information or the control logic is stored in the user memory or the program memory from where the CPU fetches the program instructions. Diagram (4)explanation (3)			

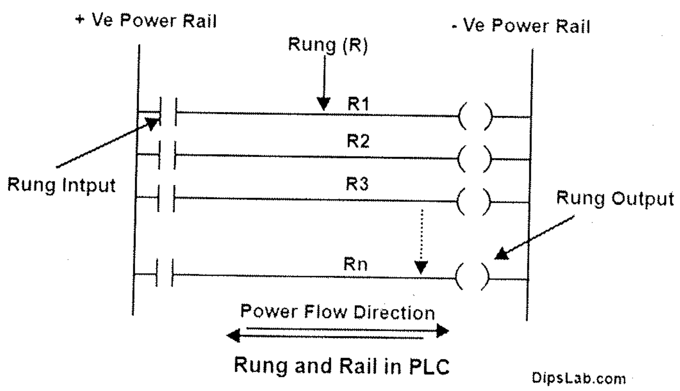
<table>
<tr><td>X</td><td>
<table>
<tr><th>#</th><th>Contents</th><th>PLC</th><th>Relay</th></tr>
<tr><td>01</td><td>Basic</td><td>Programmable Logic Control (PLC) is a solid-state computerized industrial controller that performs software logic by using input &amp; output modules, CPU, memory, and others.</td><td>Relay is an electro-mechanical switching hardware device (Hardware Switching Device).</td></tr>
<tr><td>02</td><td>Function</td><td>PLC plays a monitoring as well as controlling role in designing circuits.</td><td>Relay plays only a controlling role the designing circuit. Monitoring not so easy with a relay.</td></tr>
<tr><td>03</td><td>Working</td><td>In the PLC, we can write the program using different types of programming languages.</td><td>In the Relay, we cannot write the program.</td></tr>
<tr><td>04</td><td>Operation (Digital &amp; Analog Signals)</td><td>PLC is operated on the digital system.</td><td>Relay is operated on the analog system.</td></tr>
<tr><td>05</td><td>Function</td><td>PLC consists of more programming functions like timer, counter, memory, etc.</td><td>Relay gives only one fault detecti function. And it does not have much-advanced functionalities.</td></tr>
<tr><td>06</td><td>Design</td><td>You can easily modify the designing circuit.</td><td>Modification of the electronic circuit is more difficult as compar to PLC.</td></tr>
<tr><td>07</td><td>I/O</td><td>PLC has more capabilities of input and output modules.</td><td>The relay does not have more capabilities.</td></tr>
<tr><td>08</td><td>Flexibility</td><td>PLC provides more flexibility than the relay.</td><td>The relay provides less flexibility.</td></tr>
<tr><td>09</td><td>Fault</td><td>You can easily find the fault by using the software.</td><td>It is very hard to find fault in the Relay circuit.</td></tr>
</table>
<i>Refer Appendix</i></td><td>7*1</td><td>7</td><td></td></tr>
<tr><td>X1</td><td>Star delta starter
Refer Appendix
<i>Variations in symbols and structure of program can be accepted , as different PLCs has different programming methods</i></td><td></td><td>7</td><td></td></tr>
<tr><td>X11</td><td>Refer Appendix
<i>Variations in symbols and structure of program can be accepted , as different PLCs has different programming methods</i></td><td></td><td></td><td></td></tr>
<tr><td>XIII</td><td></td><td>7*1</td><td>7</td><td></td></tr>
</table>

Mnemonic	Name	Symbol	Description
ADD	Add		Adds Source A to Source B and stores the result in the Destination.
SUB	Subtract	 SUB Subtract Source A ? ?? Source B ? ?? Dest ? ??	Subtracts Source B from Source A and places the result in the Destination.
MUL	Multiply	 MUL Multiply Source A ? ?? Source B ? ?? Dest ? ??	Multiplies Source A by Source B and stores the result in the destination.
DIV	Divide	 DIV Divide Source A ? ?? Source B ? ?? Dest ? ??	Divides Source A by Source B and places the result in the Destination.
SQR	Square Root	 SQR Square Root Source A ? ?? Source B ? ?? Dest ? ??	Calculates the square root of the source and places the integer result in the Destination.
NEG	Negate		Changes the sign (+, -) of the Source and stores the result in the Destination.
ABS	Absolute		Takes the absolute value of the Source an places the result in the Destination.

Refer Appendix

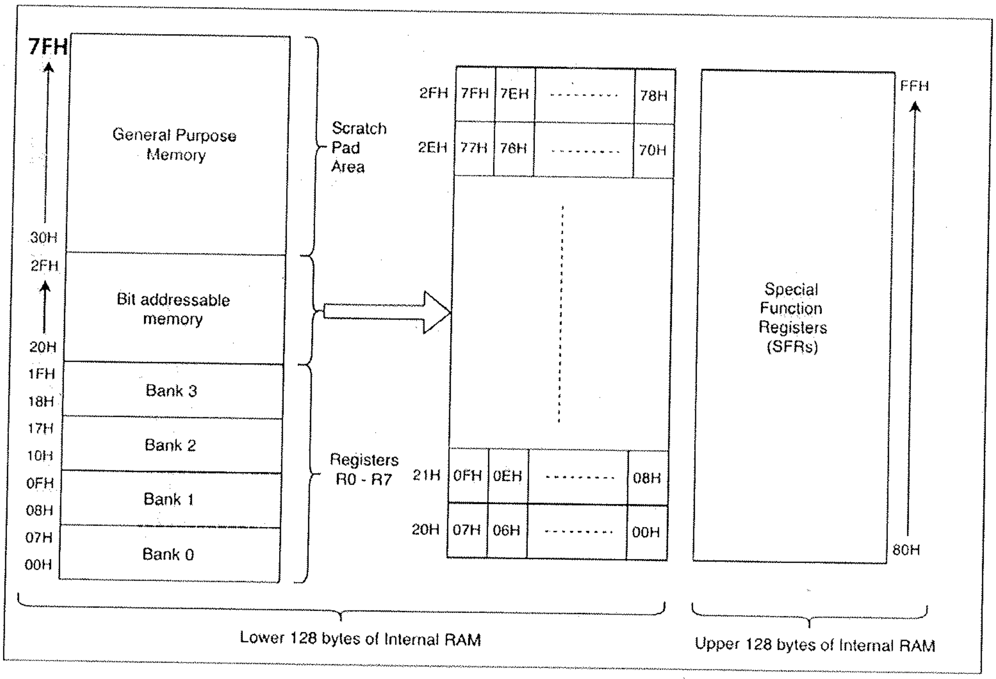
Variations in symbols can be accepted

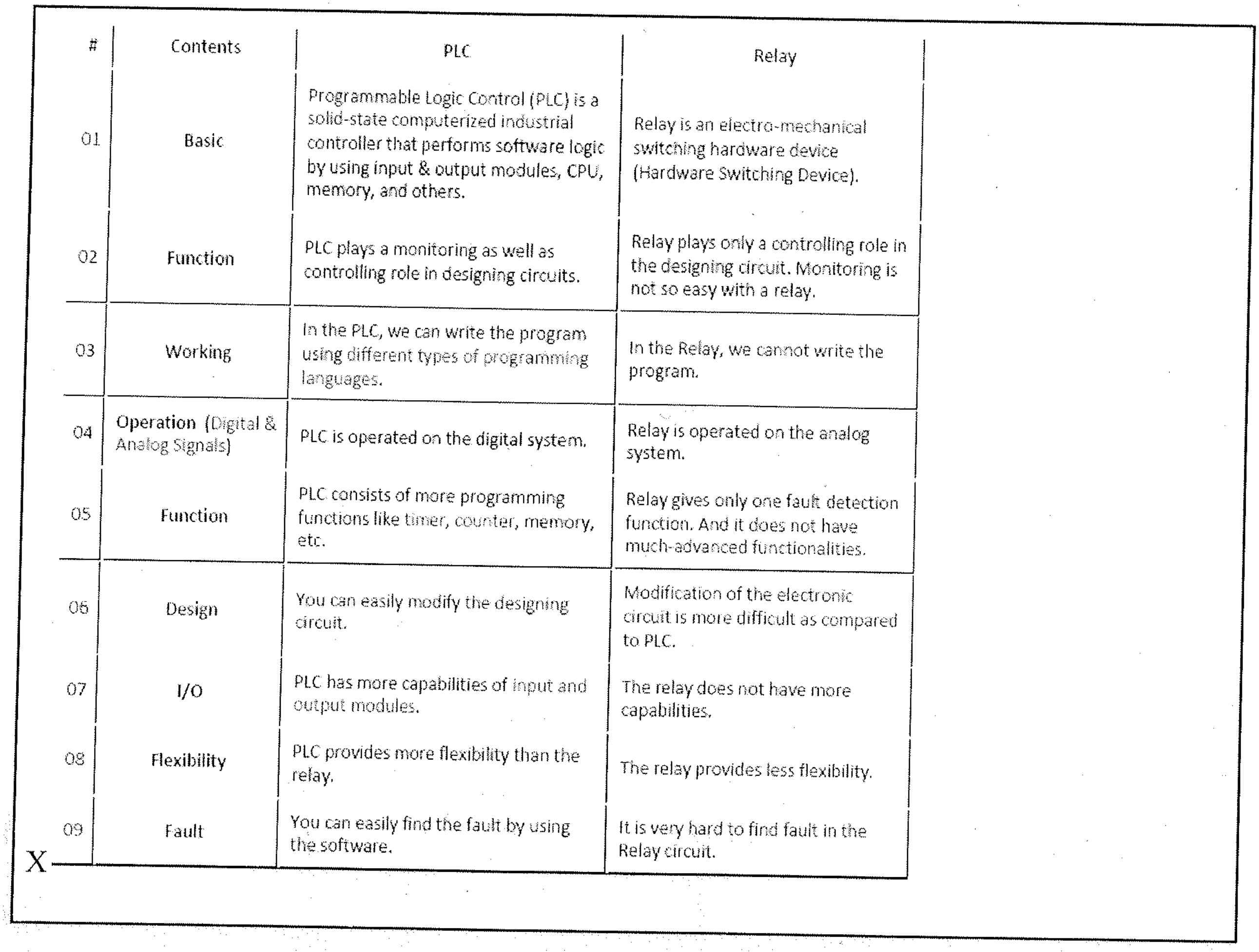
XIV	 PLC Programming Languages Ladder Diagram (LD) Instruction List (IL) Structured Text (ST) Function Block Diagram (FBD) Sequential Function Charts (SFC) DipsLab.com	4+3	7	

	 Fig + explanation(4) LISTING(3)			

APPENDIX

IV

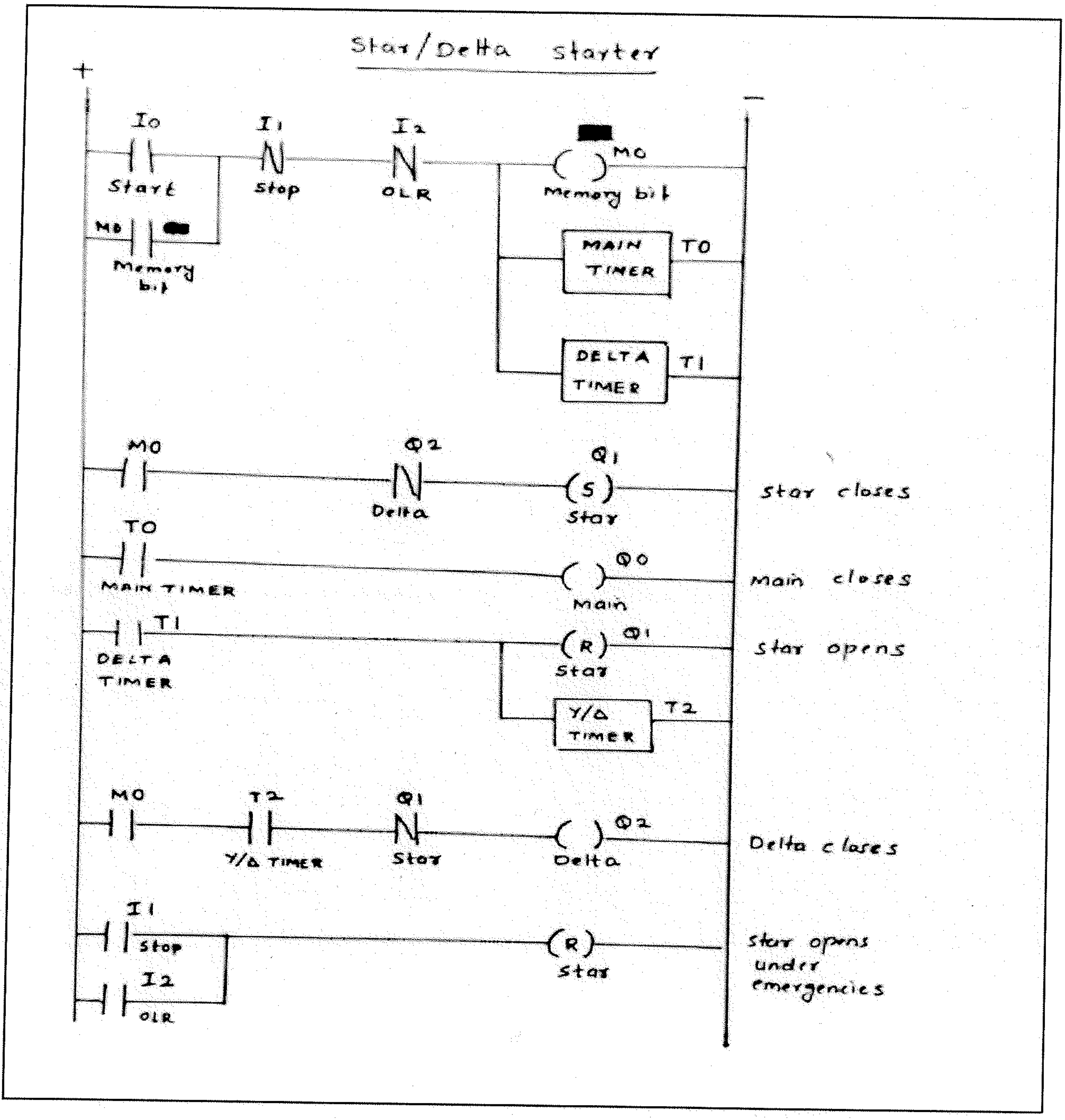




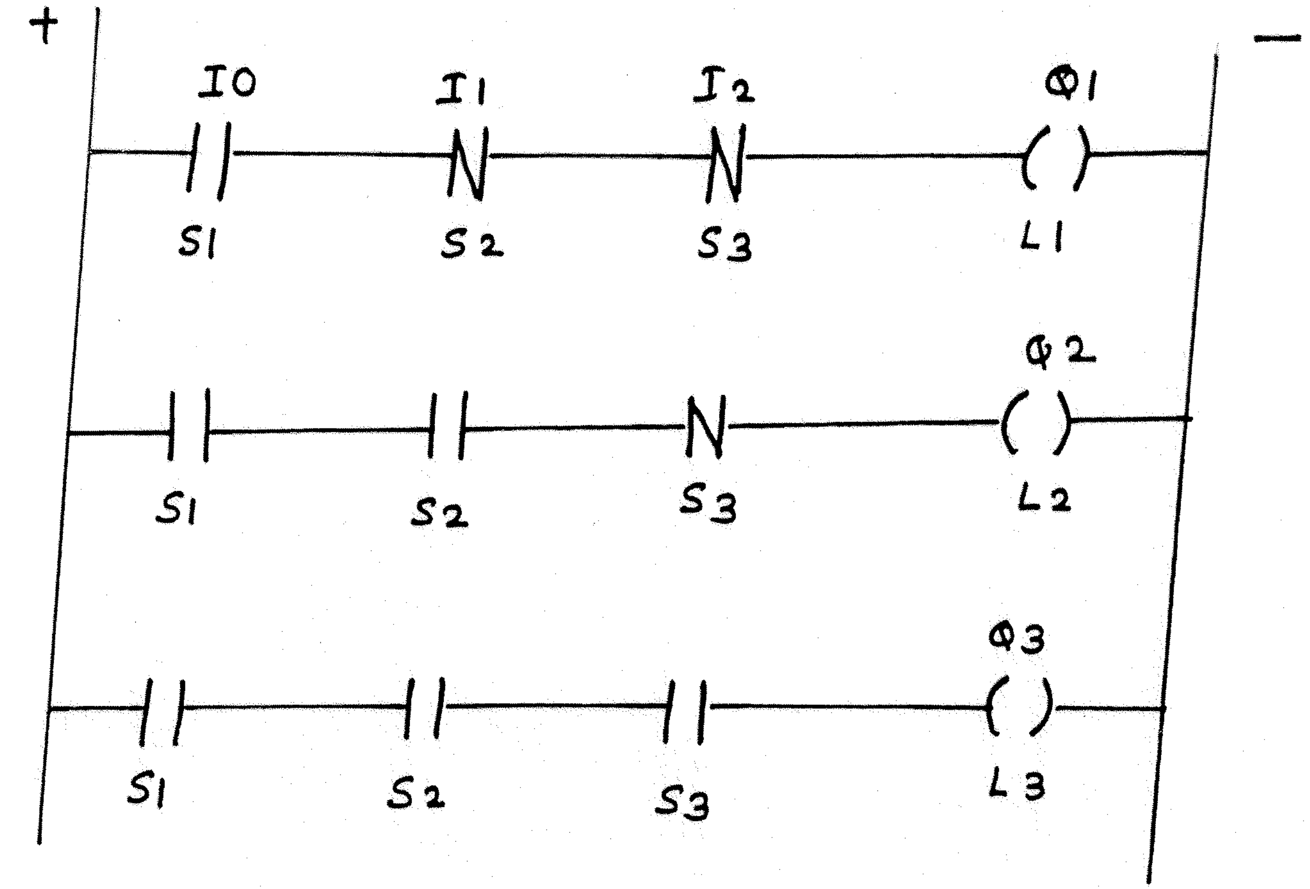
#	Contents	PLC	Relay
01	Basic	Programmable Logic Control (PLC) is a solid-state computerized industrial controller that performs software logic by using input & output modules, CPU, memory, and others.	Relay is an electro-mechanical switching hardware device (Hardware Switching Device).
02	Function	PLC plays a monitoring as well as controlling role in designing circuits.	Relay plays only a controlling role in the designing circuit. Monitoring is not so easy with a relay.
03	Working	In the PLC, we can write the program using different types of programming languages.	In the Relay, we cannot write the program.
04	Operation (Digital & Analog Signals)	PLC is operated on the digital system.	Relay is operated on the analog system.
05	Function	PLC consists of more programming functions like timer, counter, memory, etc.	Relay gives only one fault detection function. And it does not have much-advanced functionalities.
06	Design	You can easily modify the designing circuit.	Modification of the electronic circuit is more difficult as compared to PLC.
07	I/O	PLC has more capabilities of input and output modules.	The relay does not have more capabilities.
08	Flexibility	PLC provides more flexibility than the relay.	The relay provides less flexibility.
09	Fault	You can easily find the fault by using the software.	It is very hard to find fault in the Relay circuit.

Appendix

X1

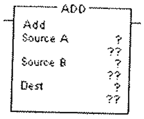
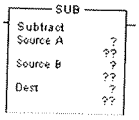
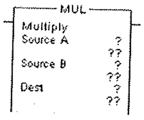
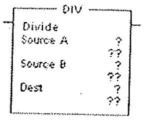
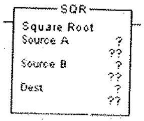
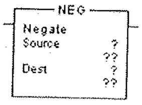
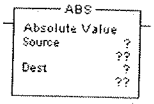

Star/Delta starter
+
−
I0
Start
I1
Stop
I2
OLR
[redacted]
M0
Memory bit
M0
[redacted]
Memory bit
MAIN TIMER
T0
DELTA TIMER
T1
M0
Q2
Delta
Q1
(S)
Star
star closes
T0
MAIN TIMER
Q0
Main
main closes
T1
DELTA TIMER
(R)
Q1
Star
star opens
Y/Δ TIMER
T2
M0
T2
Y/Δ TIMER
Q1
Star
Q2
Delta
Delta closes
I1
Stop
I2
OLR
(R)
star
star opens under emergencies

X11



GO DOWN WIRING

XIII

Mnemonic	Name	Symbol	Description
ADD	Add		Adds Source A to Source B and stores the result in the Destination.
SUB	Subtract	 SUB Subtract Source A ? ?? Source B ? ?? Dest ? ??	Subtracts Source B from Source A and places the result in the Destination.
MUL	Multiply	 MUL Multiply Source A ? ?? Source B ? ?? Dest ? ??	Multiplies Source A by Source B and stores the result in the destination.
DIV	Divide	 DIV Divide Source A ? ?? Source B ? ?? Dest ? ??	Divides Source A by Source B and places the result in the Destination.
SQR	Square Root	 SQR Square Root Source A ? ?? Source B ? ?? Dest ? ??	Calculates the square root of the source and places the integer result in the Destination.
NEG	Negate		Changes the sign (+, -) of the Source and stores the result in the Destination.
ABS	Absolute		Takes the absolute value of the Source and places the result in the Destination.