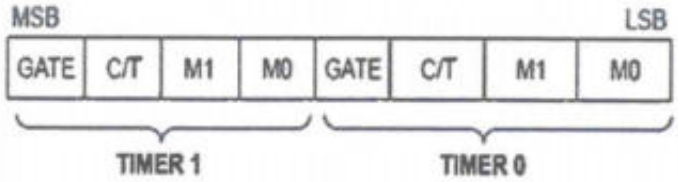
## SCORING INDICATORS (VERSION B )

**COURSE NAME : MICROCONTROLLER &PLC**

**COURSE CODE : 6031C** **QID :2102240043**

| Q No | Scoring Indicators | Split score | Sub Total | Total Score |
|---|---|---|---|---|
| | **PART A** | | | 9 |
| I. 1 | No of I/O Ports = 4 | 1 | 1 | |
| I. 2 | IE (Interrupt Enable) Register,<br>IP (Interrupt Priority) Register<br>*( any one)* | 1 | 1 | |
| I. 3 | PUSH<br>POP | 0.5*2 | 1 | |
| I. 4 | CLR A | 1 | 1 | |
| I. 5 | **Disadvantages of PLC**<br>1. There's too much work required in connecting wires.<br>2. There's difficulty with changes or replacements.<br>3. It's always difficult to find errors<br>4. It requires a skillful workforce.<br>5. When a problem occurs, hold-up time is indefinite, usually long.<br>***(write any two)*** | 0.5*2 | 1 | |
| I. 6 | Lights,<br>Alarms<br>Actuators<br>Solenoid Valves<br>Contactors,<br>Motors<br>Relays<br>***(write any two)*** | 0.5*2 | 1 | |

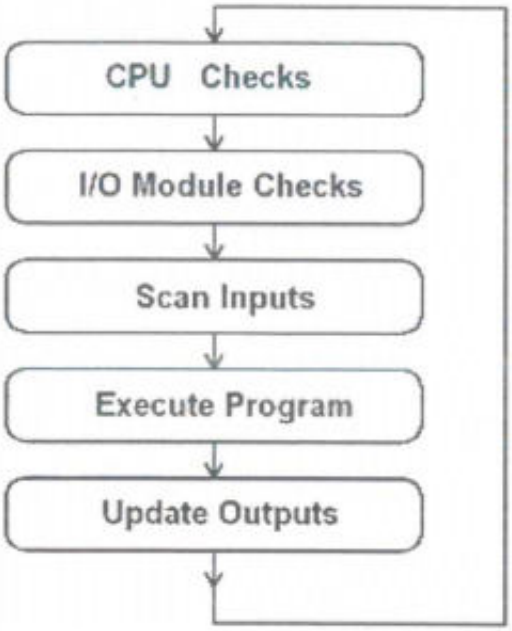
| | | | | |
|---|---|---|---|---|
| I.7 | Industrial Applications of PLC<br>• The Transportation System likes Conveyor Belt System.<br>• Packing and Labeling System in Food & Beverage.<br>• Automatic Bottle or Liquid Filling System.<br>• Packaging and Labeling System in Pharma Industries.<br>***(any two applications)*** | 0.5*2 | 1 | |
| I. 8 | Power rails | 1 | 1 | |
| I. 9 |  <br>*(Similar diagrams can be accepted)* | 1 | 1 | |

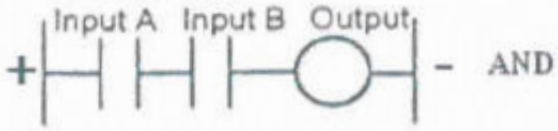
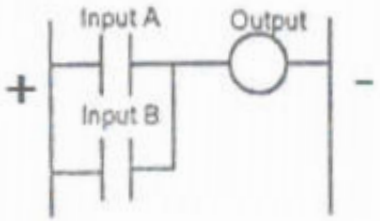
| Q No | Scoring Indicators | Split score | Sub Total | Total Score |
|---|---|---|---|---|
| | **PART B** | | | 30 |
| II. 1 |  <br>• **Gate:** when gate is set to 1 , it Enable Timer/Counter only when the INT0/INT1 pin is high and TR0/TR1 is set. | 3 | 3 | |

<table>
<tr><td></td><td>
<ul>
<li>C/T: If this bit is equal to one then it acts as a counter or if it is 0 then it act as a timer.</li>
<li>M1, M0: Decides the mode of the timer</li>
</ul>
Bit format-2<br>label-1
</td><td></td><td></td><td></td></tr>
<tr><td>II. 2</td><td>(See appendix for block diagram of 8051)</td><td>3</td><td>3</td><td></td></tr>
<tr><td>II. 3</td><td>Interrupts are the events that interrupts the normal operation of the 8051 microcontroller..<br><br>5 Interrupts in 8051 ( with order of normal priority)<br><br>1. External hardware interrupts INTO<br>2. Timer 0 overflow interrupt T0<br>3. External hardware interrupts INT1<br>4. Timer 1 overflow interrupt T1<br>5. Serial interrupt ( TI & RI)<br><br>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<br><br>Definition -1<br>Listing-2</td><td>3</td><td>3</td><td></td></tr>
<tr><td>II. 4</td><td><u>Alternate functions of port 3 pins</u><br><br>
<table>
<tr><th>P3 bit</th><th>function</th><th>Description</th></tr>
<tr><td>P3.0</td><td>RXD</td><td>Serial Data Reception</td></tr>
<tr><td>P3.1</td><td>TXD</td><td>Serial Data Transmission</td></tr>
<tr><td>P3.2</td><td>INTO</td><td>External Interrupt 0</td></tr>
<tr><td>P3.3</td><td>INT1</td><td>External Interrupt 1</td></tr>
<tr><td>P3.4</td><td>TO</td><td>Timer 0</td></tr>
<tr><td>P3.5</td><td>T1</td><td>Timer 1</td></tr>
<tr><td>P3.6</td><td>WR</td><td>Write Signal</td></tr>
</table>
</td><td>1*3</td><td>3</td><td></td></tr>
</table>

| | | | | |
|---|---|---|---|---|
| | P3.7 \| *RD* \| Read Signal | | | |
| II. 5 | **XCH** -Exchanges the value of the Accumulator with the value contained in register/Memory location.<br>Eg:- if [A]=30H and R6=[45]H, after<br>XCH A, R6; [A]=45H and [R6]=30H<br>**SWAP** -interchanges lower order(bit 3 to 0) and higher order nibbles(bit 7 to 4) of the accumulator.<br>Eg: if [A]=78H after<br>SWAP A; [A]=87H | 1.5*2 | 3 | |
| II. 6 | **Control Word of 8255 programmable peripheral interface**<br>*(See appendix)* | 3 | 3 | |
| II.7 | List general features of PLC (any three Each carry 1 mark)<br>• Processor Module: Single microprocessor device<br>• Architecture:Programmable Logic Controllers (PLC) was designed to replace the relay-based systems<br>• Programmable Logic Controllers are generally Specified by<br>  ○ No of inputs available<br>  ○ No of output available<br>  ○ Types of input available - analog/digital<br>  ○ types of power supply -ac/dc<br>  ○ programming language<br>• Programming language -There are programming languages for PLC including ladder diagram, SFC, STL, ST and so on. | 0.5*6 | 3 | |

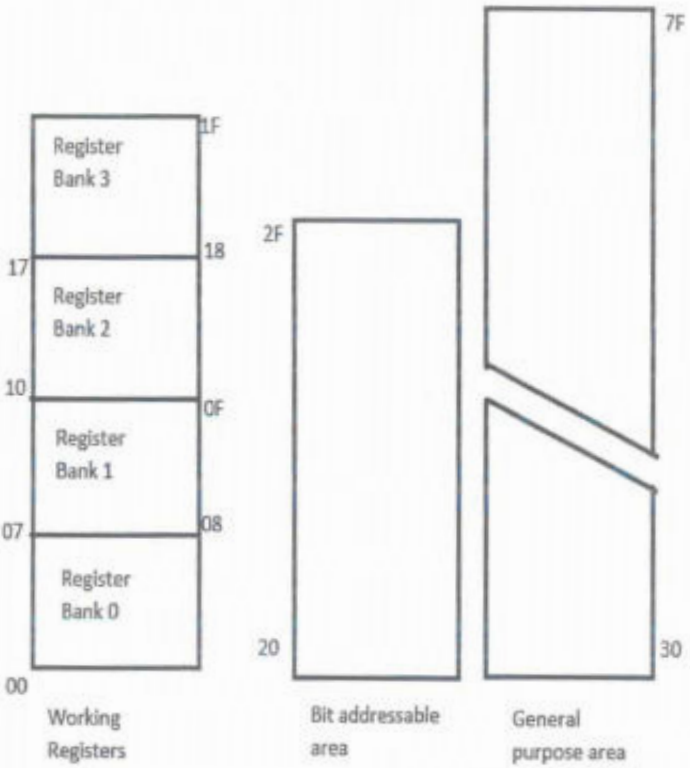
| | | | | |
|---|---|---|---|---|
| | • Other general features<br>  ○ High reliability.<br>  ○ Strong anti-interference quality<br>  ○ Good flexibility.<br>  ○ Easy-Operation<br><br>*(Write any six features)* | | | |
| II. 8 | **Operation of PLC**<br><br>PLCs operate by continually scanning programs and repeat this process many times per second.<br><br>When a PLC starts, it runs checks on the hardware and software for faults, also called a self-test.<br><br>If there are no problems, then the PLC will start the scan cycle.<br><br>The scan cycle consists of three steps: input scan, executing program(s), and output scan.<br><br>Input Scan : PLC reads the status of the inputs and solves the logic.<br><br>Execute Program (or Logic Execution) : The PLC executes a program one instruction at a time<br><br>Output Scan : When the ladder scan completes, the PLC updates the status of the outputs | 3 | 3 | |

| | | | | |
|---|---|---|---|---|
| |  <br>*Diagram-2*<br>*Explanation-1* | | | |
| II. 9 | **Factors considered while selecting PLC**<br>• Type of I/O's<br>• Memory and Programming Requirements<br>• Compact or Modular PLC<br>• Instruction Set/CPU<br>• PLC Scan Time<br>• Sinking & Sourcing PLC<br>• Manufacturer's Support and Backup<br>***(any six points)*** | 0.5*6 | 3 | |

| | | | | |
|---|---|---|---|---|
| II. 10 |  AND<br>  OR<br>***(Each carries 1.5 marks)***<br>*(Variation in symbolic representation can be accepted)* | 1.5*2 | 3 | |

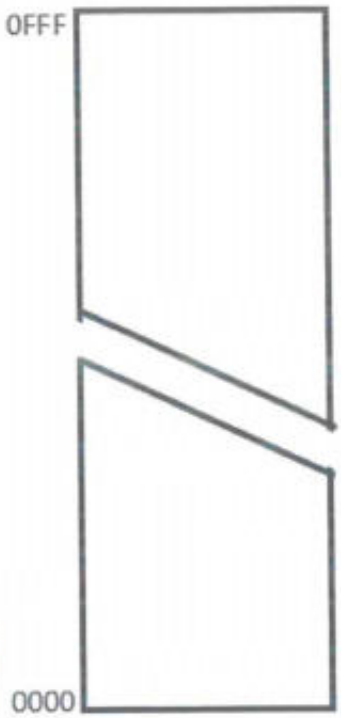
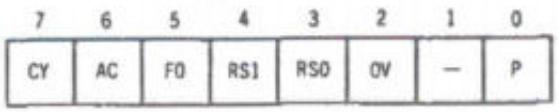
| Q No | Scoring Indicators | Split score | Sub Total | Total Score |
|---|---|---|---|---|
| | **PART C** | | | 84 |
| III | **<u>Internal memory organisation of 8051 microcontroller</u>**<br><br>Internal Data Memory (RAM)<br><br>The Data Memory or RAM of the 8051 Microcontroller stores temporary data and intermediate results that are generated and used during the normal operation of the microcontroller. the 8051 Microcontroller have 128B of RAM. i.e., memory addresses from 00H to 7FH and are divided into Working Registers (organized as Register Banks), Bit – Addressable Area and General Purpose RAM | 4+3 | 7 | |

In the first 128B of RAM (from 00H to 7FH), the first 32B i.e., memory from addresses 00H to 1FH consists of 32 Working Registers that are organized as four banks with 8 Registers in each Bank.




Internal programme memory (ROM)

Program Memory (ROM) is used for a permanent saving program being executed, In 8051 Microcontroller, the code or instructions to be executed are stored in the Program Memory, which is also called as the ROM of the Microcontroller. The original 8051 Microcontroller by Intel has 4KB of internal ROM.

| | | | | |
|---|---|---|---|---|
| |  <br>**Block representation - 4**<br>**Explanation - 3** | | | |
| IV | **Bit format of PSW**<br>PSW Program Status Word Register<br>  <br>The program status word (PSW) register is an 8-bit register. It is also referred to as the flag register.<br>• CY: Carry flag. This flag is set whenever there is a carry out from the D7 bit after an 8 bit addition or subtraction.<br>• AC: Auxiliary carry flag If there is a carry from D3 and D4 during an ADD or SUB operation, this bit is set;<br>• F0: Available to the user for general purposes. | 3+4 | 7 | |

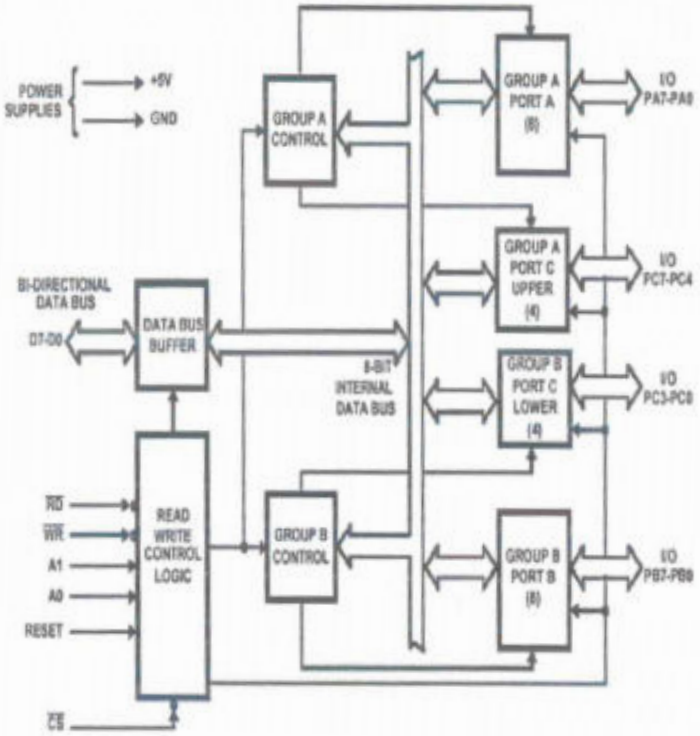
- RS0, RS1: Register bank selects bits These two bits are used to select one of the four register banks from internal RAM

| RS1 | RS0 | Reg Bank Selected |
|---|---|---|
| 0 | 0 | Bank 0 |
| 0 | 1 | Bank 1 |
| 1 | 0 | Bank 2 |
| 1 | 1 | Bank 3 |

- OV: Overflow flag: This flag is set whenever the result of a signed number operation is too large, causing the high- order bit to overflow into the sign bit.
- P: Parity flag The parity flag reflects the number of 1s in the A (accumulator) register only. If the A register contains an odd number of 1s, then P=1.
  P=0 if A has an even number of 1s.

***structure - 3***
***explanation - 4***

| | | | | |
|---|---|---|---|---|
| V | **8255 Block diagram** | 5+2 | 7 | |




8255 is a programmable I/O device that acts as an interface between peripheral devices and the microprocessor for parallel data transfer.

The three ports are Port A, Port B and Port C and as each port has 8 lines, but the 8 bits of port C is divided into 2 groups of 4-bit each.

These are given as port C lower i.e., PC3 – PC0 and port C upper i.e., PC7 – PC4. And are arranged in group of 12 pins each thus designated as Group A and Group B.

***Block Diagram:5***

***Explanation:2***

| | | | | |
|---|---|---|---|---|
| VI | **<u>Addressing modes</u>**<br>**Immediate addressing mode**<br>**Register addressing mode**<br>**Direct addressing mode**<br>**Register indirect addressing Mode** | 5+2 | 7 | |

**Indexed addressing mode**
**Implied Addressing Mode**

**Immediate addressing mode:**

The data is provided in the instruction itself. The data is provided immediately after the opcode.

Example:
MOV R3, #45H; move data #45H to R3

**Register addressing mode**:

The source and the destination both are registers, and must be of the same size. The data transfer can take place between Rn (n=0 to 7) registers and the Accumulator (A) only and cannot be done between Rn registers.

Example:
MOV A, R5; move contents of R5 to Accumulator

**Direct addressing mode:**

The source or destination address is specified by using 8-bit data in the instruction. Only the internal data memory can be used in this mode.

Example:
MOV 80H, R6; move contents of R6 to RAM Location 80H

MOV R2, 45H; move contents ofRAM location 45H to register R2

**Register indirect addressing Mode**

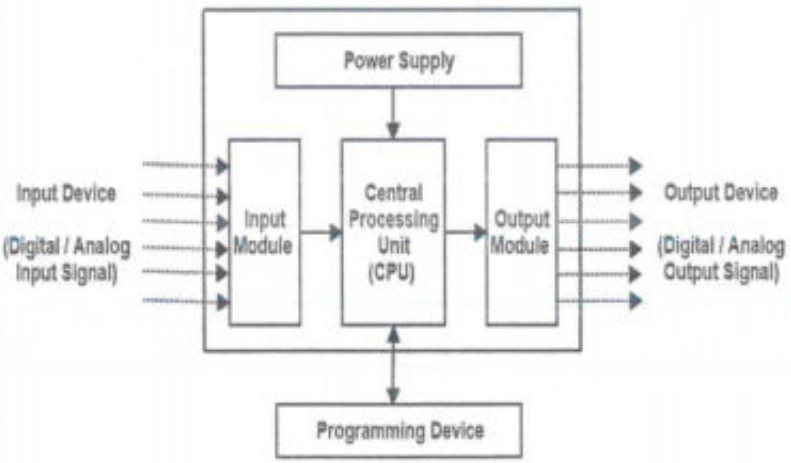
In this mode, the source or destination address is given in the register. By using register indirect addressing mode, the internal or external addresses can be accessed. Example:

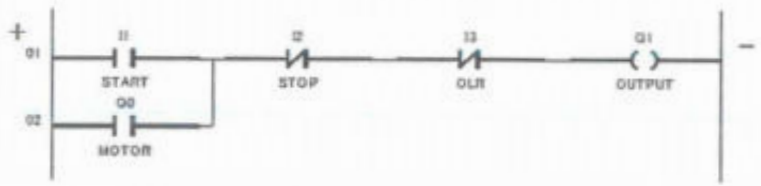
MOV A, @R0H; move contents of RAM location whose address is held by R0 into A

| | | | | |
|---|---|---|---|---|
| | **Indexed addressing mode**<br><br>Indexed addressing mode allows accessing memory locations by adding an offset to a base address.<br>The offset is usually stored in one of the registers.Indexed addressing mode of 8051 microcontroller is the sum of two registers.<br>Examples:-<br>MOVC A, @A+PC;<br>MOVC A, @A+DPTR<br><br>**Implied Addressing Mode**<br>In the implied addressing mode, there will be a single operand. These types of instruction can work on specific registers only. These types of instructions are also known as register specific instructions.<br>Examples:-<br>RL A;<br><br>*(any three)*<br>*Listing-1*<br>*Explanation-4*<br>*Example-2* | | | |
| VII | Assembly language programme to add R0 and R1 and store in external memory location 2500 H<br><br>ORG 4000<br>4000 MOV A, R0 ; copy 1st No. to A<br>ADD A, R1 ; 1st No. + 2nd No.<br>MOV DPTR, #2500H<br>MOVX @DPTR,A ; Save Sum<br>HERE : SJMP HERE<br>END<br><br>Accumulator = | 7 | 7 | |

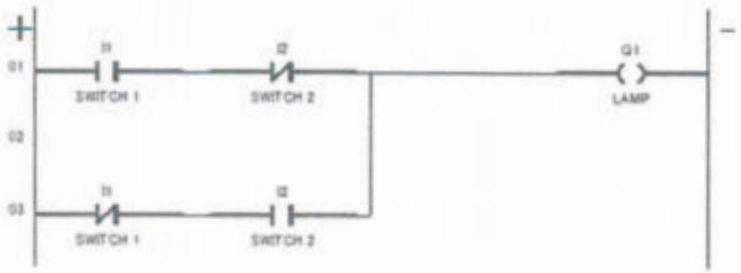
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

| | | | | |
|---|---|---|---|---|
| | Carry flag = 0 | | | |
| VIII | <table><tr><td>MOV A, #5AH</td><td>[A]=5A</td></tr><tr><td>RR A</td><td>[A]=2D</td></tr><tr><td>SWAP A</td><td>[A]=D2</td></tr><tr><td>CLR C</td><td>CY=0</td></tr><tr><td>RRC A</td><td>[A]=69 CY=0</td></tr><tr><td>CPL A</td><td>[A]=96</td></tr><tr><td>RL A</td><td>[A]=2D</td></tr></table> | 1*7 | 7 | |
| IX | <table><tr><td></td><td>PLC</td><td>Relay</td></tr><tr><td>1)Basic</td><td>It is a solid-state computerized industrial controller that performs software logic by using input & output modules, CPU, memory, and others.</td><td>Relay is an electro-mechanical switching hardware device (Hardware Switching Device)</td></tr><tr><td>2)Function</td><td>It plays a monitoring as well as controlling role in designing circuits.</td><td>It plays only a controlling role in the designing circuit. Monitoring is not so easy with a relay.</td></tr><tr><td>3) Working:</td><td>In PLC, we can write the program using different types of programming languages.</td><td>In Relay, we cannot write the program.</td></tr></table> | 1*7 | 7 | |

| | | | | | |
|---|---|---|---|---|---|
| | 4) Design: | Easily modify the designing circuit. | Modification of the electronic circuit is more difficult as compared to PLC | | |
| | 5)Memory | consist of memory to store programme | no memory | | |
| | 6) I/O | It has more capabilities of input and output modules. | The relay does not have more capabilities. | | |
| | 7) Operation: | PLC is operated on the digital system. | It is operated on the analog system. | | |
| | ***(any 7 comparisons- each carries 1 mark)*** | | | | |
| X | **Block diagram of PLC**<br>  <br><br>**Processor section (CPU):** The processor section is the brain of PLC which consists of RAM, ROM, logic solver and user memory. The central processing unit is the heart of PLC. | | 4+3 | 7 | |

| | | | | |
|---|---|---|---|---|
| | **Input and output module:** The input module is a mediator between input devices and central processing unit (CPU) which is used to convert analog signal into digital signal.<br><br>The output module is a mediator between output devices and the central processing unit (CPU) which converts digital signal into analog signal.<br><br>**Power supply:** power supply is provided to the processor unit, input and output module unit. Power supply may be an integral or separately mounted unit.<br><br>**Memory section:** The memory section is the area of the CPU in which data and information is stored and retrieved.<br><br>**Programming device:** Programming devices are dedicated devices used for loading the user program into the program memory or edit it and to monitor the execution of the program of the PLC.<br><br>***Block diagram : 4***<br><br>***Explanation : 3*** | | | |
| XI | a) **<u>Direct online Starter</u>**<br><br>  <br><br><br>Inputs:<br><br>START- To start the motor- NO<br>STOP - To stop the motor-NC<br>OLR- Overload Relay<br>Output:<br><br>MOTOR- Motor | 3.5+<br><br>3.5 | 7 | |

b) **Staircase lighting control**



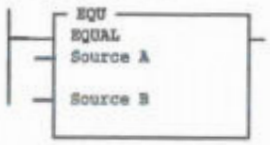

Lamp conditions :

| Switch 1 | Switch 2 | Lamp |
|---|---|---|
| ON | OFF | ON |
| ON | ON | OFF |
| OFF | ON | ON |
| OFF | OFF | OFF |

**(or similar ladder logic diagram )**
*Direct online Starter-3.5*

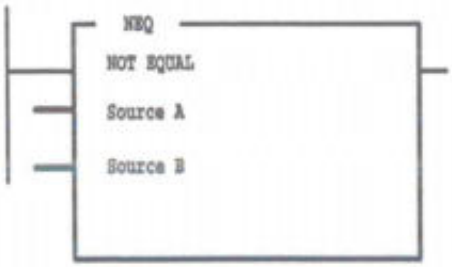
*Staircase lighting-3.5*

*(Variations in symbolic representation can be accepted)*

| | | | | |
|---|---|---|---|---|
| XII | Star-delta starter<br>*(See appendix for diagram)* | 7 | 7 | |
| XIII | **Comparison instructions:**<br>• EQU (equal) Instruction<br>  <br> | 7 | 7 | |

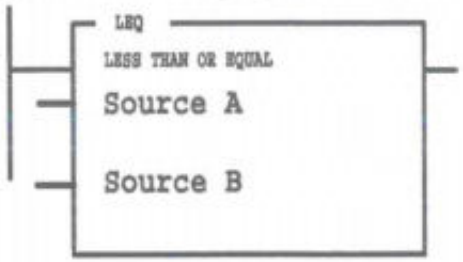
Use the EQU instruction to test whether two values are equal. If source A and source B are equal, the instruction is logically true.

- NEQ (not equal) Instruction



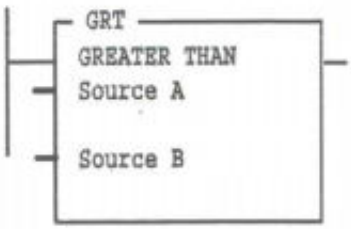

Use the NEQ instruction to test whether two values are not equal. If source A and source B are not equal, the instruction is logically true.

- LEQ (Less than or equal) Instruction

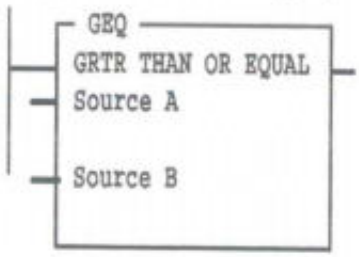



Use the LEQ instruction to test whether one value (source A) is less than or equal to another (source B). If the value at source A is less than or equal to the value at source B, the instruction is logically true.

- GRT(Greater than) Instruction




Use the GRT instruction to test whether one value

| | | | | |
|---|---|---|---|---|
| | (source A) is greater than another (source B). If the value at source A is greater than the value at source B, the instruction is logically true.<br><br>• GEQ(Greater than or equal) Instruction<br>  <br><br><br>Use the GEQ instruction to test whether one value (source A) is greater than or equal to another (source B).If the value at source A is greater than or equal to the value at source B, the instruction is logically true.<br><br>*(any 4 comparison instructions)*<br>*(Variations in symbolic representation can be accepted)* | | | |
| XIV | (i) **Normally open contact**<br><br>  <br><br>The NO (Normally open)instruction looks and operates like a normally open relay contact. There is a memory bit associated with each NO instruction that is linked to the status of an input device or internal condition.<br>If the input device is ON or Closed, then the corresponding bit in the data memory is set to 1, thus allowing the current to flow from its left side to its right-hand side.Otherwise, it is set to 0, thus blocking the current. | 7 | 7 | |

(ii) **Normally closed contact**

The NC(Normally closed) instruction looks and operates like a normally closed relay contact. There is a memory bit associated with each NC instruction that is linked to the status of an input device or internal condition.
If the input device is OFF or Open, then the corresponding bit in the data memory is set to 1, thus allowing the current to flow from its left side to its right-hand side.Otherwise, it is set to 0, thus blocking the current.

(iii) **OFF delay timer**
A off-delay (TOF) timer is a PLC programming instruction which use to switch off the output or system after a certain amount of time.

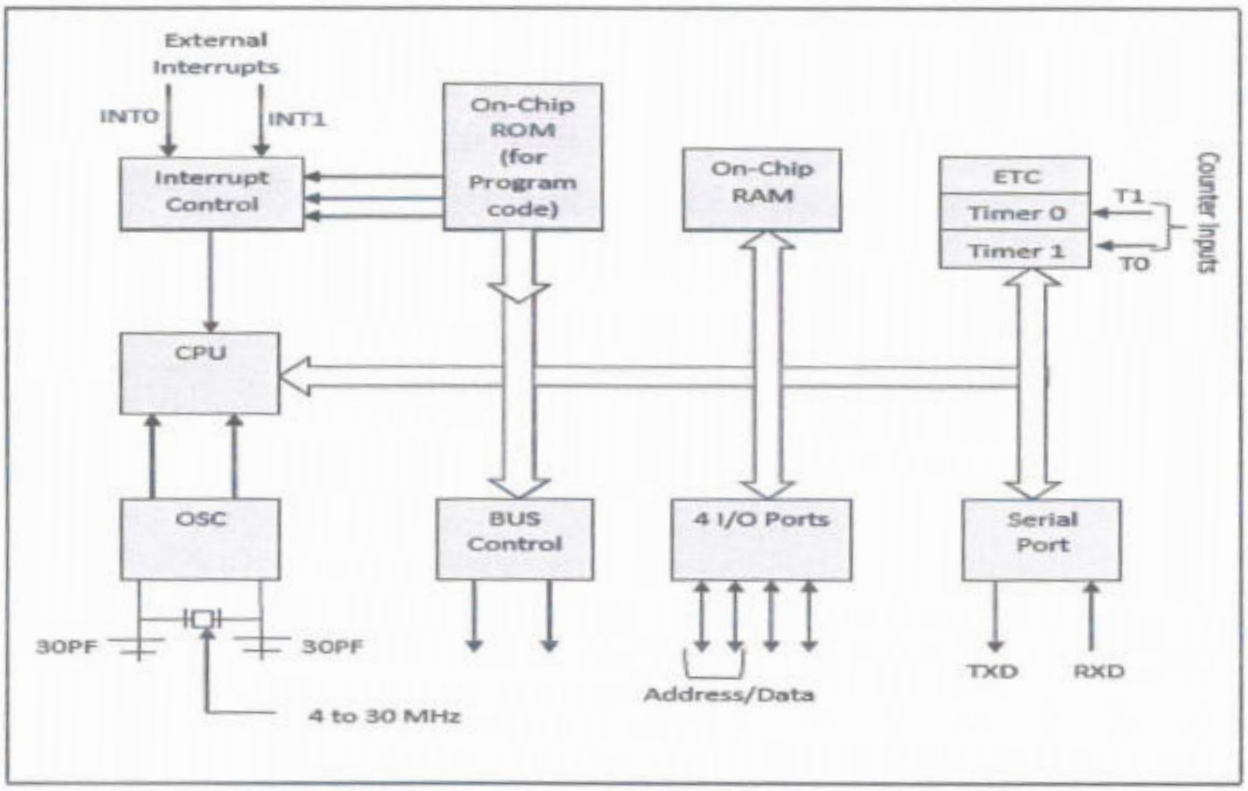
(iv) **Output relay coil**

If the condition of the left link of the output relay coil is ON then the corresponding bit in the output data memory is set. The device wired to this output is also energized.

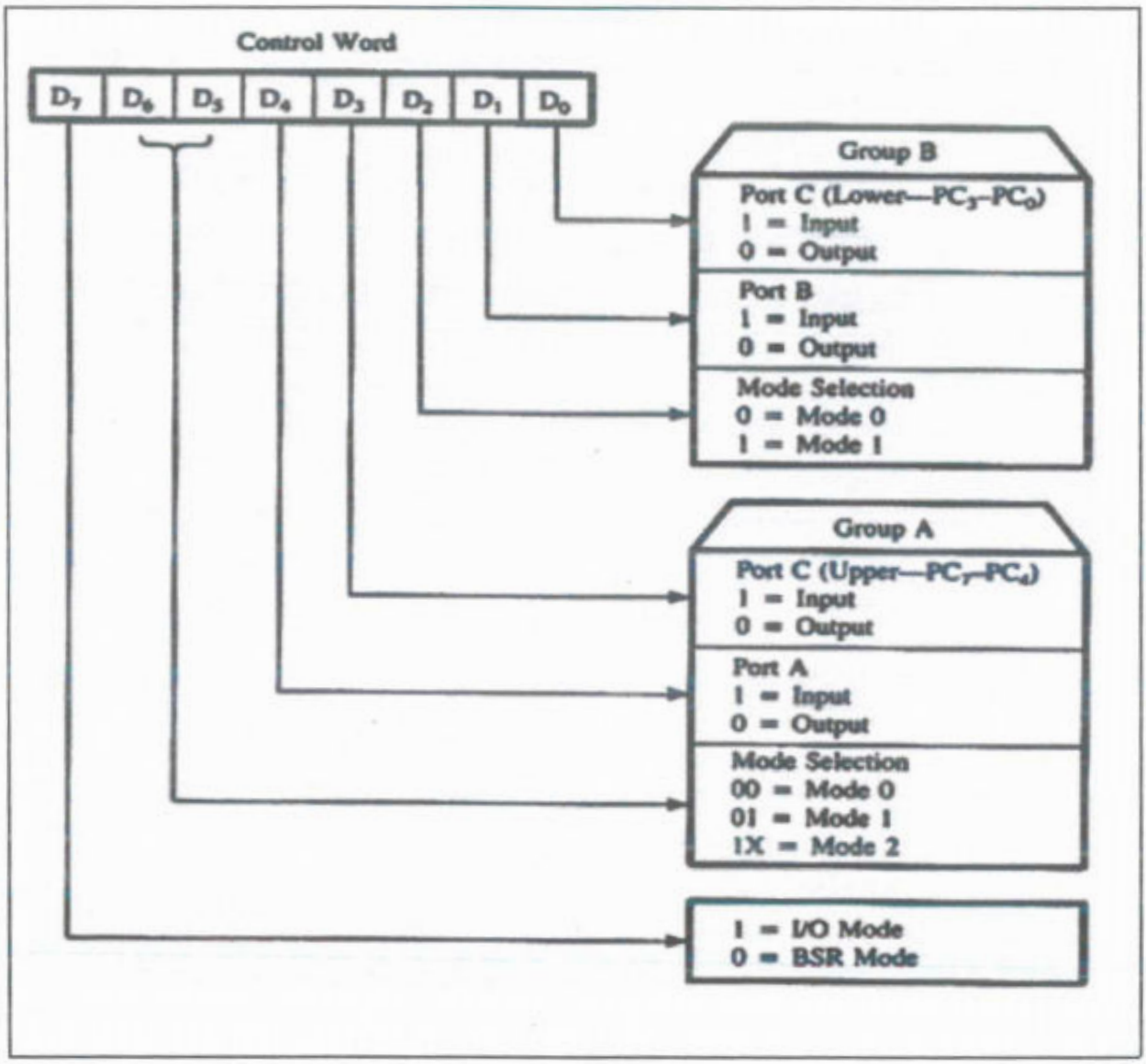
# Appendix

## Part B

### II.2 Schematic block diagram of 8051

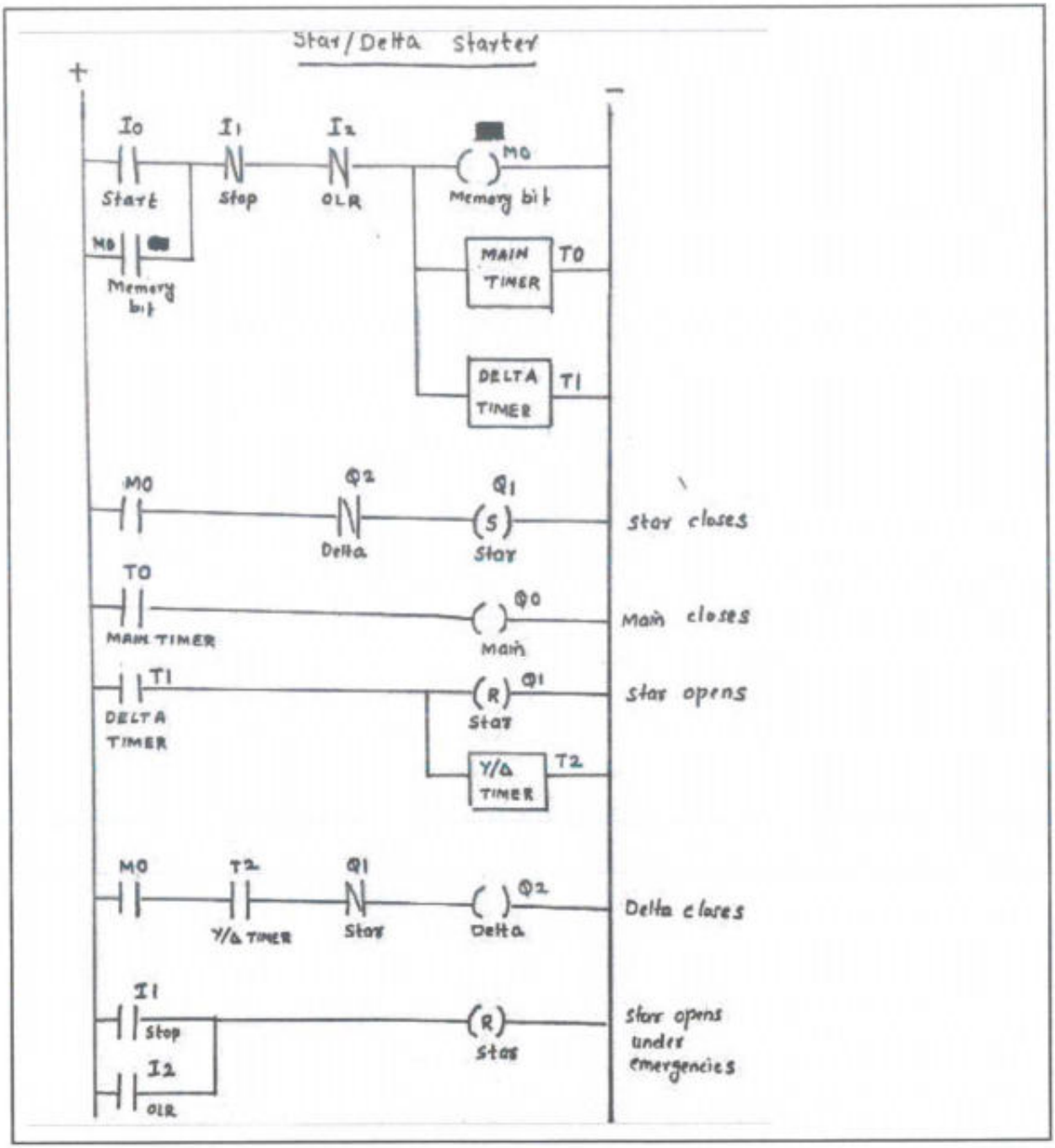


II.6


Control Word
D7
D6
D5
D4
D3
D2
D1
D0
Group B
Port C (Lower—PC3–PC0)
1 = Input
0 = Output
Port B
1 = Input
0 = Output
Mode Selection
0 = Mode 0
1 = Mode 1
Group A
Port C (Upper—PC7–PC4)
1 = Input
0 = Output
Port A
1 = Input
0 = Output
Mode Selection
00 = Mode 0
01 = Mode 1
1X = Mode 2
1 = I/O Mode
0 = BSR Mode

## Part C

**XII.**




**ladder logic or similar**