Chapter-I

Short type questions [2 marks each]

Q-1) **What do you mean by Full Custom IC Technology?**
[S-16(Q1-a)], [S-19(Q1-a)]

Ans:- Full Custom IC technology is a methodology for designing IC by specifying the layout of each individual transistor and the interconnection between them. It minimizes the interconnection length, size, signal transmission and routing wires of transistors.

Q-2) **Write any two characteristic of Embedded System.**
[S-17(Q1-a)]

Ans:- Embedded system has the following characteristics-
- → Single functioned
- → Tightly constrained
- → Reactive and real time

Q-3) **What IC Technology?**
[S-17(Q2-a)]

Ans:- IC technology is the design technology which involves in which manner a digital(gate-level)implementation is mapped onto an IC. IC technology is independent of processor technology, all type of processor can be mapped to any type of IC technology.

Q-4) **Define design technology.**
[S-17(Q3-a)]

Ans:- Design technologyis the study, design, development, application, implementation, support and management for the express purpose of communicating product design intent and constructability. It involves the manner in which we convert our concept of desired functionality into an implementation.

Medium type questions [5 marks each]

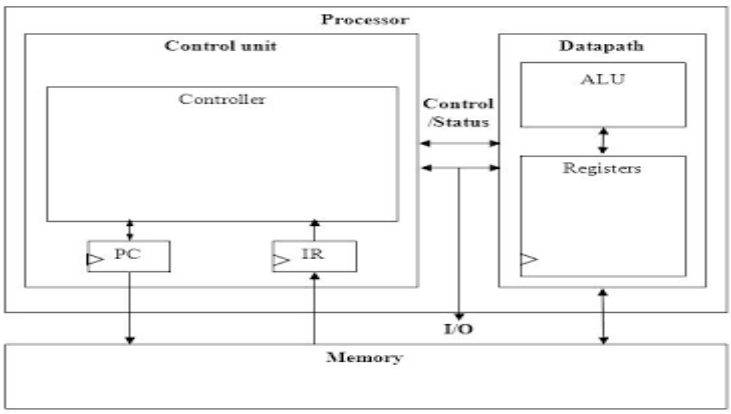
Q-1) **Define embedded system and list some applications of Embedded system.**
[S-16(Q1-b)], [S-19(Q6-a)]

Ans:- An embedded system is a system that has embedded software and computer hardware, which makes it a system dedicated for an application or part of an application or product or a part of larger system.
A short list of embedded system-
- → Antilock brakes
- → Automatic Teller Machine (ATM)
- → Automatic toll system
- → Life support system
- → Digital camera
- → Cruise control
- → TV set-top box
- → Home security system
- → Electronic card reader
- → Fax machines
- → Finger print identifiers
- → Satellite phones
- → Photo copiers
- → Teleconferencing systems

Q-2) **What is processor technology? Explain briefly the general purpose processor technology. [S-17(Q1-b)]**

Ans:- Processor technology relates to the architecture of the computation engine used to implement a system's desired functionality.

General Purpose Processor (software):-

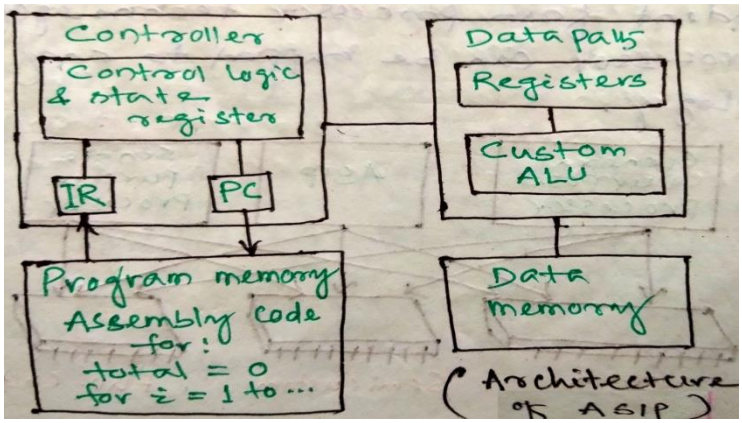


The designer of a GPP or microprocessor builds a programmable device that is suitable for a variety of applications to maximize the number of devices sold. One feature of such a processor does not know what program will run on the processor, so the program can not built into the digital ckt. Another feature is a general data path-the data path must be general enough to handle a variety of computations that has a large register file and one or more general purpose ALUs. A designer simply uses the GPP by programming the processor's memory to carry out the required functionality. This part of an implementation is referred as the Software portion.

Long type questions [7 marks each]

Q-1) **Explain briefly about application specific purpose processor with giving suitable example. [S-16(Q1-c)], [S-19(Q1-b)]**

Ans:-



Processor technology:-
An application-specific instruction set processor (ASIP) is a component used in system-on a chip design. The instruction set of an ASIP is tailored to benefit a specific application. It can serve a compromise between the other processor options.

An ASIP is a programmable processor optimized for a particular class of operations having common characteristics, such as embedded control, digital signal processing or can optimize the data path for the application class, perhaps adding special functional units for common operations and eliminating other infrequently used units.

Using ASIP in an embedded system can be provide the benefit of flexibility while still achieving good performance, power and size. However, such processors can require large NRE cost to build the processor itself and to build a compiler, if these items don't already exist. Example- Microcontroller and DSP.

Q-1) **Discuss the types of embedded system technology. [S-19(Q1-c)]**

Ans:- The technologies that are central to embedded system are- Processor technology, IC technology & Design technology.

Processor technology:- Processor technology relates to the architecture of the computation engine used to implement a system's desired functionality. Although the term processor is usually associated with programmable software processor. It is of 5 types- GPP(Software), SPP(Hardware), ASIP, Microcontroller & DSP.

IC technology:-Every processor eventually be implemented on an IC. IC technology involves the manner in which a digital (gate-level) implementation is mapped onto an IC. IC technology is independent from processor technology; any processor can be mapped to any IC technology. IC technology differences can be recognised by the semiconductor's layers. The bottom layer from the transistors, the middle layer form the logic components and the upper layer connect these components with wires. It is of 3 types- Full Custom/VLSI, Semicustom/Gate array & Standard cell and PLD.

Design technology:- Design technology involves the manner in which convert the concept of desired system functionality into an implementation. It refines the system through several abstraction levels. At the system level, the desired functionality is described in some language like C; which is called system specification. This specification is refined by distributing portions of it among several general and/or single purpose processor, yielding behavioural specification for each processor. These specifications then refined into register-transfer specification by converting behaviour on GPP to assembly code and by converting behaviour on SPP to a connection of register transfer components and state machines. Then the R-T level specification for a SPP assembly into a logic specification consisting of Boolean equation. Finally, the remaining specifications refined into a implementation consisting of machine code for GPP and a gate level net list for SPP.

Chapter-II

Short type questions [2 marks each]

Q-1) **Write down any two difference between Microprocessor and Microcontroller.[S-16(Q2-a)]**
Ans:-**Microprocessor:-**

- It is CPU on a single chip. It contains ALU, GPR, SP, PC, Clock timing ckt and interrupts ckt.
- It has many instructions to move data between memory & CPU.

Microcontroller:-

- Microcontroller contains the circuitry of μP and in addition it has built in ROM, RAM, Timers, I/O devices and counters.
- It has one or two instructions to move data between memory and CPU.

Q-2) **Define Microcontroller.[S-19(Q2-a)]**
Ans:- A microcontroller is a small computer on a single MOS IC. It is a compact IC designed to govern a specific operation in an embedded system. A typical microcontroller includes a processor, memory and input/output (I/O) peripherals on a single chip.

Q-3) **What is $\overline{\text{PSEN}}$?[S-19(Q5-a)]**
Ans:- **$\overline{\text{PSEN}}$:-** It is program store enable. It is output control signal. It is a read strobe to external program memory. This goes low during external program memory accesses.

Q-4) **What are the various ports available in 8051? [S-19(Q4-a)]**
Ans:- The 8051 microcontroller has four ports as Port-0, Port-1, Port-2 and Port-3;each port is 8-bit in single chip made.

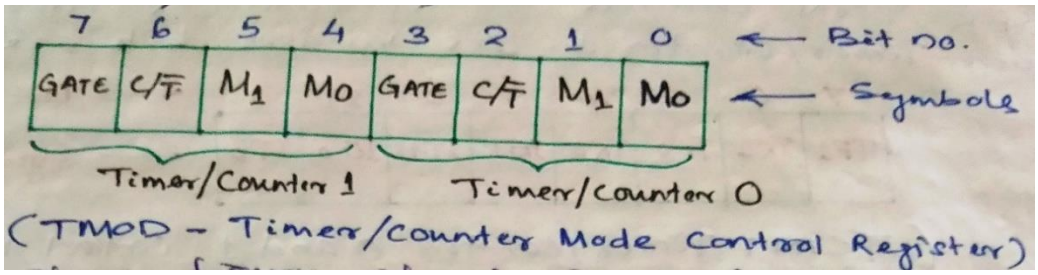
Medium type questions [5 marks each]

Q-1) **Write down the difference between Microcontroller and Microprocessor. [S-17(Q6-b)]**
Ans:-

Microprocessors		Microcontrollers
1	It is only a general purpose computer CPU	It is a micro computer itself
2	Memory, I/O ports, timers, interrupts are not available inside the chip	All are integrated inside the microcontroller chip
3	This must have many additional digital components to perform its operation	Can function as a micro computer without any additional components.
4	Systems become bulkier and expensive.	Make the system simple, economic and compact
5	Not capable for handling Boolean functions	Handling Boolean functions
6	Higher accessing time required	Low accessing time
7	Very few pins are programmable	Most of the pins are programmable
8	Very few number of bit handling instructions	Many bit handling instructions
9	Widely Used in modern PC and laptops	widely in small control systems
E.g.	INTEL 8086,INTEL Pentium series	INTEL8051,89960,PIC16F877

Q-2) **Explain briefly TMOD register with example.[S-19(Q4-b)]**

Ans:-TMOD register controls the modes of operation of the timer/counter 0and timer/counter 1.



The above figure shows the various bits of TMOD register.

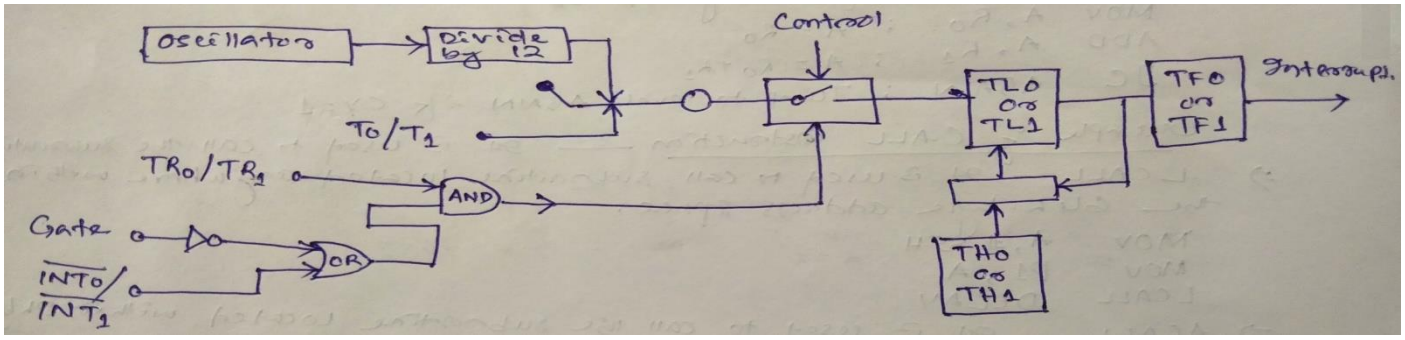
M1	M0	Mode	Operation
0	0	Mode-0	13-bit counter/timer
0	1	Mode-1	16-bit counter/timer
1	0	Mode-2	8-bit counter/timer with automatic reload
1	1	Mode-3	T/C-0 operates in two separate T/C and T/C-1 does not work

C/$\overline{T}$:- Timer or counter selector. When C/$\overline{T}$ =1, counter operation (i/p from pins T_0& T_1). When C/$\overline{T}$ =0, timer operation (i/p from internal system bus)

Gate:- When Gate=1, Timer/counter-X is enabled, only if $\overline{INTX}$ pin is high and TR_X is set (x= 0 or 1)

When Gate=0, Timer/counter-X is enabled, whenever TR_X is set(x=0 or 1).

Example:- Timer/Counter in Mode-2 operation



This mode is same for both Timer/counter- 0 or 1. In this mode, T/C register TL_0 or TL_1 is made an 8-bit counter with automatic reload. The overflow from TL_0/TL_1 sets TF_0/TF_1 and it also reloads TL_0/TL_1 with contents of TH_0/TH_1 which preset by software.

Long type questions [7 marks each]

Q-1) **Give a briefly discussion about Interrupt priority register and interrupt enable register. [S-19(Q2-c)]**

Ans:- **IE (Interrupt Enable) Register**

This register is responsible for enabling and disabling the interrupt. EA register is set to one for enabling interrupts and set to 0 for disabling the interrupts. Its bit sequence and their meanings are shown in the following figure.

EA	-	-	ES	ET1	EX1	ET0	EX0

EA	IE.7	It disables all interrupts. When EA = 0 no interrupt will be acknowledged and EA = 1 enables the interrupt individually.
-	IE.6	Reserved for future use.
-	IE.5	Reserved for future use.
ES	IE.4	Enables/disables serial port interrupt.
ET1	IE.3	Enables/disables timer1 overflow interrupt.
EX1	IE.2	Enables/disables external interrupt1.
ET0	IE.1	Enables/disables timer0 overflow interrupt.
EX0	IE.0	Enables/disables external interrupt0.

IP (Interrupt Priority) Register

We can change the priority levels of the interrupts by changing the corresponding bit in the Interrupt Priority (IP) register as shown in the following figure.

- A low priority interrupt can only be interrupted by the high priority interrupt, but not interrupted by another low priority interrupt.
- If two interrupts of different priority levels are received simultaneously, the request of higher priority level is served.
- If the requests of the same priority levels are received simultaneously, then the internal polling sequence determines which request is to be serviced.

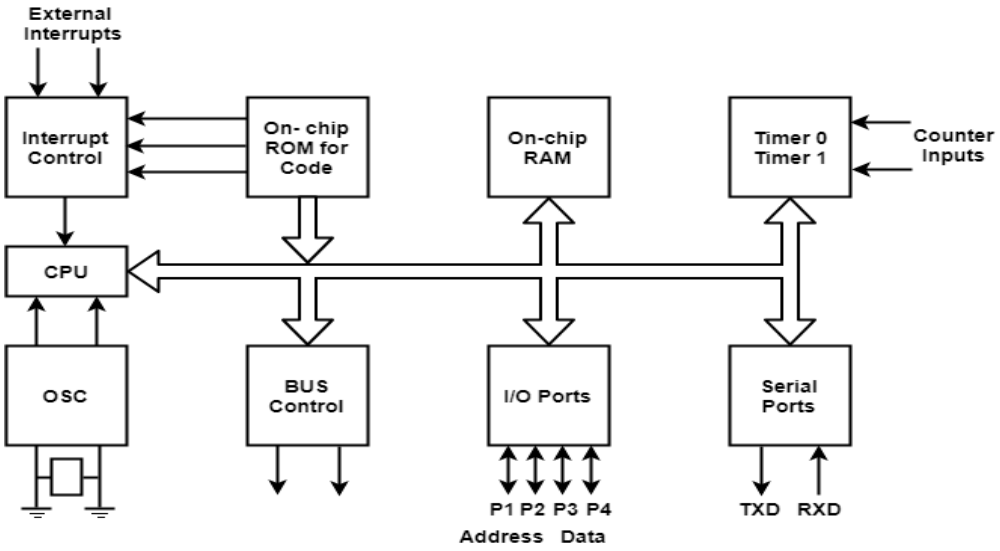
-	-	PT2	PS	PT1	PX1	PT0	PX0
bit7	bit6	bit5	bit4	bit3	bit2	bit1	

-	IP.6	Reserved for future use.
-	IP.5	Reserved for future use.
PS	IP.4	It defines the serial port interrupt priority level.

PT1	IP.3	It defines the timer interrupt of 1 priority.
PX1	IP.2	It defines the external interrupt priority level.
PT0	IP.1	It defines the timer0 interrupt priority level.
PX0	IP.0	It defines the external interrupt of 0 priority level.

Q-2) **Describe the architecture of 8051 microcontroller. [S-19(Q3-c)]**

Ans:-A microcontroller architecture consists of a cpu, two kinds of memories, I/O ports, the mode status, data registers and random logic needed for a variety of peripheral functions.



CPU:- MCS-51/8051 consists of 8-bit ALU with associated registers like A,B,PSW,SP,the 16-bit PC and DPTR register. The ALU perform arithmatic and logic operation on 8-bit operand.

BOOLEAN PROCESSOR:- There is a separate boolean processor integrated within the 8051 MC. It has own instruction sets, accumulators and bit address level RAM.It allowes bit manupulation perform operation line, comliment bit, set bit, clear bit.

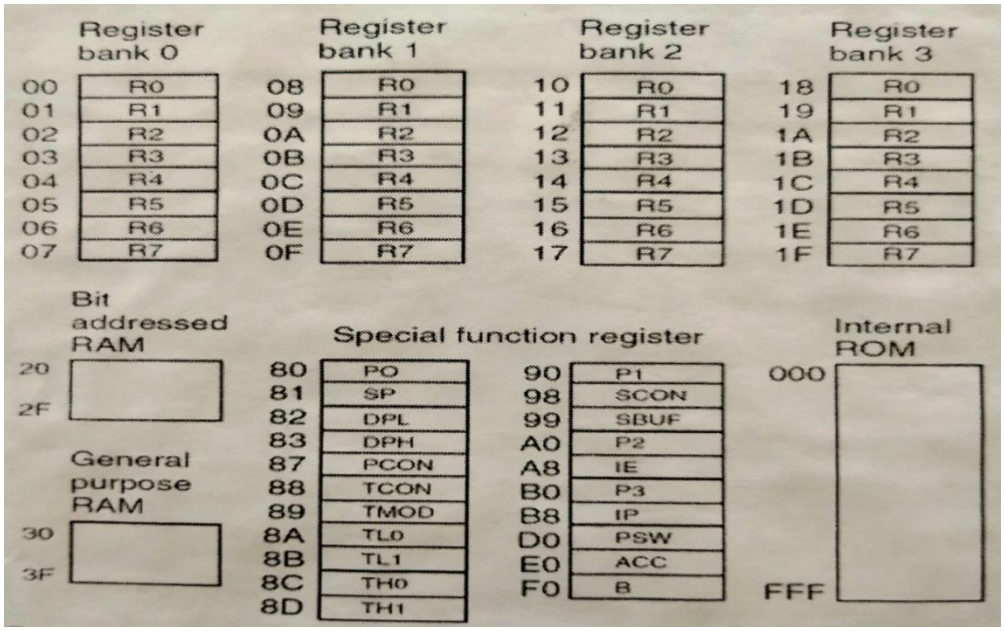
REGISTERS:- There are special function registers(SFR) which are the prgoram status word, accumulator, register, stack pointer, register for serial I/O ports, interrupt handlers.

PROGRAM & DATA MEMORY:- There are two separate program and data memory. The code is stored in ROM/EPROM. RAM of 8051 MC is 1128 bytes. RAM is used for controlling the operation of the peripheral timer/counter serial ports interrupt etc.

PORTS(P0,P1,P2,P3):- 8051 MC has four ports P_0,P_1,P_2& P_3each port is 8-bit in single chip made. There are two timer and serial interface(SI).

OSCILLATOR:- The 8051 MC used and external crystal oscillator function. The frequency of operation can be depending upon the individual device data sheets of the device can be reffered to see the operating fequency supported by typical device.

Q-3) **Discuss briefly about 8051 programming model. [S-19(Q5-c)]**
Ans:-



The programming model of 8051 is divided into two major group of registers. Registers that hold data for arithmetic & logical functions are from ALU; and registers called SFRs are from data memory. The programming model includes the working registers(WREG),flag registers (STATUS), file select registers(FSRs), multiplication register, program counter, stack and SFRs.

Working Registers (WREG):- The working register of 8051 is similar to the accumulator in other processor. It is an 8-bit register in ALU that is used in all arithmetic & logic operations. The result of arithmetic or logic operation can be stored in the WREG or in other operand register.

Bank Select Register(BSR):- The BSR is an 8-bit register but uses only the lower 4-bit to specify the data bank from 0 to F and the upper 4-bits are always 0. The data memory has 4096 registers and is divided into 16 banks, each with 256 registers.

STATUS Registers:- The Status register is an 8-bit register that uses 5 individual bits, B_0 to B_4, called flags, reflecting the data conditions of an operation; the remaining 3 bits, B_5 –B_7 are unused. The data condition flags reflect the nature of the result after an operation. The 5 flags are known as C-carry, DC-digit carry, Z-zero, OV-overflow and N-negative.

File Select Register (BSRs):- There are 3 registers, FSR_0, FSR_1 and FSR_2 hold 12-bit addresses of data registers and are used as pointers for indirect addressing. To hold a 12-bit address FSR, registers require two 8-bit registers- FSRH & FSRL and each FSR register is associated with an INDF register for indirect addressing.

Program Counter(PC):- The PC is a 21-bit register that functions as a counter and provides capability of addressing 2MB of memory. The PC consists of three 8-bit registers- PCL, PCH & PCU. The MC uses this counter to sequence the execution of the instructions.

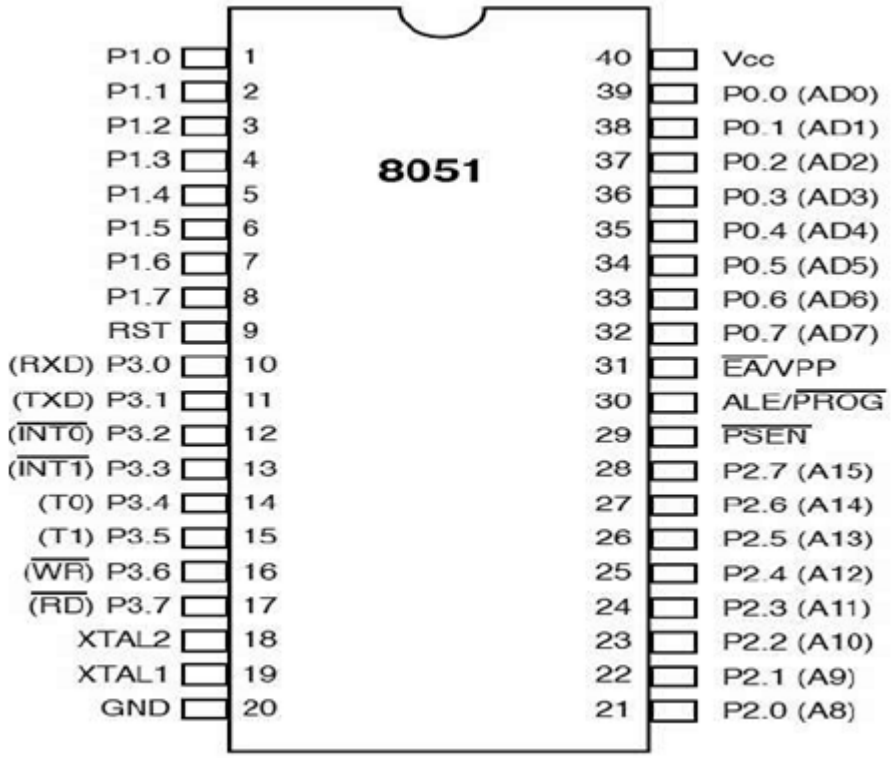
Table Pointers(TB):- These are 21-bit registers that are used as memory pointers to copy bytes between program memory and data memory.

Stack Pointer(SP):- The stack is a group of 31-word sized registers that are used for temporary storage of memory addresses during the execution of a program. The SP uses 5-bits to indicate where the instructions that are used to store and retrieve information from the 31 register stack.

Special Function Register(SFR):- In programming model, 10 blocks are shown as special function registers representing register associated with the I/O ports, support devices and process of data transfer. The SFRs associated with various devices in the MCU are I/O port, Interrupt, EERPROM, Serial I/O, Timers, CCP register, A/D converter, Synchronous serial I/O and other registers.

Q-4) **Draw the pin diagram of 8051 and explain about function of each pin. [S-19(Q3-c)]**

Ans:- The pin diagram of 8051 is shown below-



V_{cc}(Pin-40):- It is a +5V supply voltage pin.

GND(Pin-20):- It is the return pin for the power supply.

RESET(Pin-9):- The reset pin resets the 8051, only when it goes high for two or more machine cycles.

ALE/$\overline{\text{PROG}}$(Pin-30):- The address latch enable(ALE) o/p pulse indicates that the valid address bits are available on their respective pins. The ALE signal is valid only for external memory access.

$\overline{\text{PSEN}}$(Pin-29):- It is program strobe enable. It is o/p control signal which a read strobe to external program memory. This goes low during external program memory access.

$\overline{\text{EA}}/V_{PP}$(Pin-31):- It is external access. It controls the access of program of program memory. The 8051 can execute a program in external memory, only if $\overline{\text{EA}}$ is tied low. For execution of program in internal memory, the $\overline{\text{EA}}$ is tied high.

XTAL1(Pin-19):- It is i/p to the inverting amplifier which is a part of the on chip oscillator ckt. When external clock is used, it is connected to the external oscillator signal.

XTAL2(Pin-18):- It is o/p to the inverting amplifier which is a part of the on chip oscillator ckt. When external clock is used, it is left unconnected to the external oscillator signal.

PORT 0(Pin-32 to 39):-Port 0 is an 8-bit bidirectional bit addressable I/O port. This has been allotted an address in the SFR address range. Port 0 acts as multiplexed address/data lines during external memory access(i.e) when $\overline{EA}$ is low and ALE emits a valid signal. In case of controllers with on chip EPROM, Port 0 receives code bytes during programming of the internal EPROM.

PORT 1(Pin-1 to 8):- Port 1 acts as a 8-bit bidirectional bit addressable port. This has been allotted an address in the SFR address range.

PORT 2(Pin-21 to 28):- Port 2 acts as a 8-bit bidirectional bit addressable I/O port. This has been allotted an address in the SFR address range. During external memory accesses, port2 emits higher 8-bit of address(A_8 –A_{15}) which are valid, if ALE goes high and $\overline{EA}$ is low. Port2 also receives higher order address bits during programming of the on chip EPROM.

PORT 3(Pin-10 to 17):- Port 3 is an 8-bit bidirectional bit addressable I/O port. This has been allotted an address in the SFR address range. The port3 also serve the alternative functions as follows-
P3.0- Acts as serial input data pin (RxD)
P3.1- Acts as serial output data pin (TxD)
P3.2- Acts as external interrupt pin 0 ($\overline{INT0}$)
P3.3- Acts as external interrupt pin 1 ($\overline{INT1}$)
P3.4- Acts as external input or timer-0 (T0)
P3.5- Acts as external input or timer-1 (T1)
P3.6- Acts as write control signal for external data memory ($\overline{WR}$)
P3.7- Acts as read control signal for external data memory ($\overline{RD}$)

Chapter-III

Short type questions [2 marks each]

Q-1) **Give any two examples of Jump and Call Instruction.[S-19(Q3-a)]**
Ans:-**Jump:-** It is used to jump to the level specified. Examples are-

→ **JZ:-** Jump to the level specified if accumulator is zero.

```
MOV A,R0  ; A=R0

JZ OVER   ; jump to lever OVER if A=0
```

→ **JC:-** Jump to the level specified if carry is produced (CY = 1)

```
MOV A,R0  ; A = R0

ADD A,R1  ; A = R0+ R1

JC AGAIN  ; jump to lever AGAIN if CY = 1
```

CALL:- It is used to call the subroutine. Examples are-

→ **LCALL:-** It is used to call subroutine located anywhere within the 64K byte address space.

```
MOV A,#55H  ; Move the data into Accumulator

MOV P1,A    ; Move the content of Acc. to port-1

LCALL DELAY ; Call the subroutine DELAY
```

→ **ACALL:-** It is used to call subroutine located in the 2K byte.

```
MOV A,#00H  ; Move the data into Accumulator

MOV P1,A    ; Move the content of Acc. to port-1

MOV R0,#30H ;

LCALL DELAY ; Call the subroutine DELAY
```

Q-2) **Give any two examples of program branching instruction. [S-19(Q7-a)]**
Ans:- The two examples of program branching instruction are-

```
LCALL addr16; RET
```

Medium type questions [5 marks each]

Q-1) **Explain different type of Addressing modes of 8051. [S-19(Q3-b)]**

Ans:- The addressing modes of Intel 8051 are-

Register addressing, Direct addressing, Register Indirect addressing, Immediate addressing and Bas register plus Index register Indirect addressing.

Register addressing:- The 8 working registers of the selected register bank are accessed by register addressing. The last significant 3-bits of the instruction Op-code indicate which register is to be accessed. ACC, B, DPTR and CY are also accessed by register addressing.

Direct addressing:- The only method to access SFRs is direct addressing. The lower 128 bytes of internal RAM are also accessed by this addressing.

Register-Indirect addressing:- The content of either R0 and R1 is used as a pointer to access memory locations in the 256 bytes block; the lower 128 bytes of internal RAM, the upper 128 bytes of internal RAM and the lower 256 bytes of external data memory. SFRs are not accessible by register-indirect addressing. Full 64K of external data memory are accessed by 16-bit DPTR. This addressing is also useful for the execution of PUSH or POP instruction. The SP may reside anywhere in the internal RAM.

Base Register plus Index-Register Indirect addressing:- It is used to access a byte from the location whose address is the sum of a base register (DPTR or PC) and an index register. Acc is used as index register. This mode is used look up table access. Program memory address @DPTR+A or @PC+A.

Chapter-IV

Long type questions [7 marks each]

Q-1) **Write a program to multiply the data in R_5& R_6 adding the 16-bit result to R_1& R_2. [S-19(Q4-c)]**

Ans:-

```
        ORG  0000
        LJMP MAIN
        ORG  100H
MAIN:   MOV A,R5; Move the data from register R5 to accumulator
        MOV B,R6  ; Move the data from register R6 to B
        MUL  AB   ; Multiply register B with accumulator
        ADD A,@R1; Add the LSB of 16-bit result with contents of R1
        ADD B,@R2; Add the MSB of 16-bit result with contents of R2
        MOV R1,A  ; Store the LSB of result to register R1
        MOV R2,B  ; Store the MSB of result to register R2
```

Q-2) **Write a program to generate a square wave having 33% duty cycle using 8051 instruction and its use. [S-17(Q4-c)]**

Ans:-

The 33% duty cycle means the 'ON' state is half of 'OFF' state. The output is on bit 3 of port-3.

```
BACK:SETB P3.3   ; Set bit 3 of port-3
     LCALL DELAY ; Call the delay subroutine
     CLR P3.3    ; Clear bit 3 of port-3
     LCALL DELAY ; Call the delay subroutine
     LCALL DELAY ; Call the delay subroutine again
     SJMP BACK   ; Keep doing it
```

Q-2) **Write a program to generate 2KHz square waves on pin P1.0 of port 1. Using 8051 instruction set using timer/counters.[S-17(Q5-c)]**

Ans:-

Assume the XTAL frequency = 11.0592Hz, so clock time = 1.085µs

Given frequency= 2KHz;so, time period of square wave, T = 500µs

Half of time period for high & low as well= 250µs

No of cont = 250µs/1.085µs = 230

Start of count= 65536-230= 65306= C09A hex.

So, TH= C0H and TL= 9AH

Program:-

```
          MOV TMOD,#01H; Start timer 0 in mode 1
     AGAIN:MOV TL0,#9A H  ; put 9A H in timer 0 low byte
           MOV TH0,#C0 H  ; put C0 H in timer 0 high byte
           SETB TR0       ; Start the timer run control
     NEXT: JNB TF0, NEXT  ; Stay until timer flag rolls over
           CLR TR0        ; Stop timer 0
           CPL P1.0       ; Complement P1.0 pin
           CLR TF0        ; clear timer flag 0
           SJMP AGAIN     ; Reload new data in TH0 and TL0
```

Q-3) **Write a simple program for addition of 8-bit numbers located in two memory address using 8051 microcontroller. [W-17(Q3-c)]**

Ans:-

Let us add 45H and 39H stored in memory location 51H and 52H.

```
     MOV A,51H; Get 1st data in accumulator from memory location 51H
     ADD A,52H ; Add the content of memory location 52H to the content
                 of the accumulator
     MOV 53H,A ; Store the sum in memory location 53H
HERE SJMP HERE ; Jump at the same location to end program
```

Data:- 51H – 45H Result:- 53H – 7EH
52H – 39H

Chapter-V

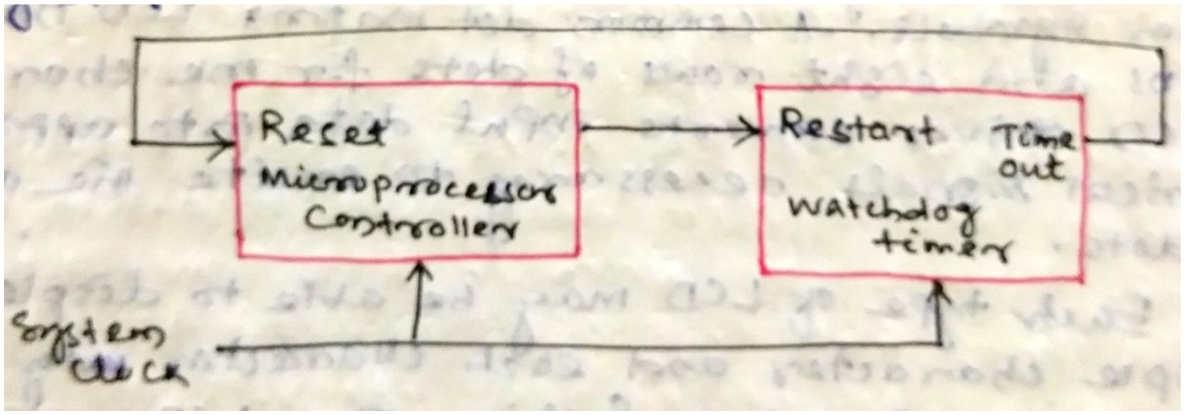
Short type questions [2 marks each]

Q-1) **Define RTC. [S-17(Q5-a)]**
Ans:- Real time clock(RTC) is a system component responsible for keeping track of time. RTC holds information like current time in 12 hour or 24 hour format, date, month, year, day of week, etc and supply timing reference to the system of power.

Q-2) **Define watch dog timer. [W-17(Q5-a)]**
Ans:-It is a timing device such that it is set for a present time interval and an event must occur during that interval else the device will generate timeout signal on failure to get that event in the watched time interval.

Medium type questions [5 marks each]

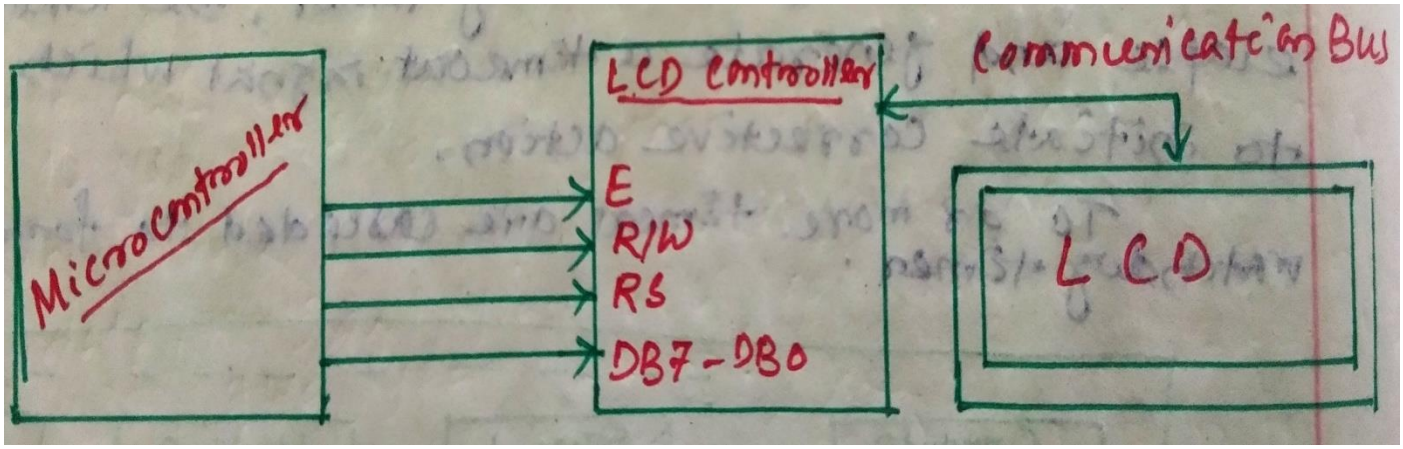
Q-1) **Explain briefly about the watch dog timer. [S-19(Q2-b)]**
Ans:-



It is a timing device such that it is set for a present time interval and an event must occur during that interval else the device will generate timeout signal on failure to get that event in the watched time interval.On that event, the watchdog timer is disabled to disable generation of timeout or reset.Timeout may result in processor start a service routine or start from beginning. A software task can also be programmed as a watchdog timer. Depending upon the internal implementation, it increments or decrements a free running counter with each clock and generates a reset signal to reset the processor; if the counter reaches zero for a down counting watchdog or the highest count value for an up counting watchdog. Simply, a watchdog timer is an electronic timer that is used to detect and recover from computer malfunctions.

An application in mobile phone is that display is off in case no GUI interaction takes place within a watched time interval. The interval is usually set at 15_S, 20_S, 25_S, 30_S in mobile phone. This saves power. Microcontroller may also provide for a watchdog timer.

Q-2) **Explain the working of LCD controller with a neat diagram. [S-19(Q5-b)]**
Ans:-



A liquid crystal display (LCD) is a low cost, low power device capable of displaying text and images. LCDs are of three types-

- Reflective LCD
- Absorption LCD
- Dot matrix LCD

Each type of LCD may be able to display multiple character and each character may be displayed in normal or inverted fashion. The LCD may permit a character to be blinking or may permit display of a curser indicating the current character. This function is difficult to implement using software. Thus an LCD controller is used to interface an LCD of 8 data i/ps and one enable i/p.

A microcontroller is connected to an LCD controller, which in turn is connected to an LCD. The LCD controller receives control words from the microcontroller, it decodes the control words and performs the corresponding actions on LCD.

Once the initialization sequence is done, control word or data can be sent to be displayed. RS is set to low to indicate that the data sent is a control word. When RS is high, this indicates that the data sent over the communication bus corresponds to a character that is to be displayed. Every time data is sent, whether it is a control word or data, the enable bit E must be toggled.

Chapter-VI

Short type questions [2 marks each]

Q-1) **What is PLC?[S-17(Q6-a)], [W-17(Q6-a)]**
Ans:-A Programmable Logic Controller or PLC is a digital computer used for automation of electromechanical process. These controllers can automate a specific process, machine function, or even an entire production line.

Q-2) **Why PLC is used for automation? [S-17(Q6-a)]**

Ans:-PLCs are used in industrial automation to increase reliability, system stability and performance, minimizing the need for human operators and the chances of human error.

Medium type questions [5 marks each]

Q-1) **Explain the internal instruction of PLC. [S-19(Q6-b)]**
Ans:- A PLC instructions are-

- Relay-type (Basic) instructions: I, O, OSR, SET, RES, T, C
- Data Handling Instructions:
- Data move Instructions: MOV, COP, FLL, TOD, FRD, DEG, RAD
- Comparison instructions: EQU (=), NEQ (≠), GEQ (≥), GRT (>).
- Mathematical instructions.
- Continuous Control Instructions (PID instructions).
- Program flow control instructions: MCR (master control reset), JMP, LBL, JSR, SBR, RET, SUS, REF
- Specific instructions:
- BSL, BSR (bit shift justify/right), SQO (sequencer output), SQC (sequencer compare), SQL (sequencer load).
- High speed counter instructions: HSC, HSL, RES, HSE
- Communication instructions: MSQ, SVC
- ASCII instructions: ABL, ACB, ACI, ACL, CAN

Internal Relays:- Auxiliary relays, markers, flags, coils, bit storage. Used to hold data, and behave like relays, being able to be switched on or off and switch other devices on or off. They do not exist as real-world switching devices but are merely bits in the storage memory.

Internal Relays Use:- In programs with multiple input conditions or arrangements. For latching a circuit and for resetting a latch circuit. Giving special built-in functions with PLCs.

Retentive relays (battery-backed relays):- Such relays retain their state of activation, even when the power supply is off. They can be used in circuits to ensure a safe shutdown of plant in the event of a power failure and so enable it to restart in an appropriate manner.

Latch Instructions (Set and Reset):- The set instruction causes the relay to self-hold, i.e. latch. It then remains in that condition until the reset instruction is received.The latch instruction is often called

a SET or OTL (output latch).The unlatch instruction is often called a RES (reset), OTU (output unlatch) or RST (reset).

Timers:- Timer is an instruction that waits a set amount of time before doing something (control time). Timers count fractions of seconds or seconds using the internal CPU clock. The time duration for which a timer has been set is termed the preset and is set in multiples of the time base used.

Data Handling Instructions:- Timers, counters and individual relays are all concerned with the handling of individual bits, i.e. single on-off signal. PLC operations involve blocks of data representing a value, such blocks being termed words.

Data handling consists of operations involving moving or transferring numeric information stored in one memory word location to another word in a different location, comparing data values and carrying out simple arithmetic operations.

4-bit register can store a positive number between 0 and +15.
8-bit: 0 and +255.
16-bit: 0 and +65535.

Data movement instructions:- There are typically 2 common instruction "sets". The single instruction is commonly called MOV (move) copies a value from one address to another.The MOV instruction needs to know 2 things:

Source – where the data we want to move is located.
Destination – the location where the data will be moved to.

We write an address here. Also, the data can be moved to the physical outputs.

Data comparison:- The data comparison instruction gets the PLC to compare two data values.Thus it might be to compare a digital value read from some input device with a second value contained in a register. PLCs generally can make comparisons for:

- less than (< or LESS),
- equal to (= or EQU),
- less than or equal to (<= or LEQ),
- greater than (> or GRT),
- greater than or equal to (>= or GEQ), and
- not equal to (NEQ).

Arithmetic (mathematical) Instructions:-

PLCs almost always include math functions to carry out some arithmetic operations:

Addition (ADD) – The capability to add one piece of data to another.
Subtraction (SUB) – The capability to subtract one piece of data from another.
Multiplication (MUL) – The capability to multiply one piece of data by another.
Division (DIV) – The capability to divide one piece of data from another.

Continuous control (PID Instruction):- Continuous control of some variable can be achieved by comparing the actual value of the variable with the desired set value and then giving an output depending on the control law required.Many PLCs provide the PID calculation to determine the controller output as a standard routine. Control instructions are

used to enable or disable a block of logic program or to move execution of a program from one place to another place.
The control instructions include:

- Master Control instruction (MC/MCR)
- Jump to label instruction (JMP)
- Label instruction (LBL)
- Jump to Subroutine instruction (JSR)
- Subroutine instruction (SBR)
- Return from Subroutine instruction (RET)
- Shift Registers

Master Control/ Master Control Reset (MC/MCR):-
When large numbers of outputs have to be controlled, it is sometimes necessary for whole sections of program to be turned on or off when certain criteria are realized. This could be achieved by including a MCR instruction. A MCR instruction is an output instruction.
The master control instruction typically is used in pairs with a master control reset. Different formats are used by different manufacturers: MC/MCR (master control/master control reset),MCS/MCR (master control set/master control reset) orMCR (master control reset).

Jump Instructions:-
The JUMP instructions allow to break the rung sequence and move the program execution from onerung to another or to a subroutine. The Jump is a controlled output instruction.You can use multiple jump to the same label.Jumps within jumps are possible, There are:
Jump to Label. 2.Jump to subroutine
RETURN / END:- A Return from Subroutine instruction marks the end of Subroutine instruction. When the rung condition of this instruction is true, it causes the PLC to resume execution in the calling program file at the rung following the Jump to Subroutine instruction in the calling program.

Q-2) **State the difference between a programmable controller and a computer. [S-19(Q7-b)]**
Ans:- The difference between a programmable logic controller & computer is follows-

PLC	Computer
Ruggedised design for industrial environments	Designed mainly for data processing and calculation
Ability to operate in high temperatures and humidity	Limited environmental range
High immunity to signal noise	Optimised for speed
Integrated proprietary command interpreter	Support for multiple development environments
Limited memory	Significant and expandable memory
Optimised for single-thread processing	Multitasking capability

Q-3) **Explain advantages of PLC. [W-17(Q6-b)]**

Ans:- The advantages of PLC are as follows:

1. **Flexible in Nature:** One model of PLC can be used for different operations as per requirement.
2. **Easy to install and troubleshooting:** In hard wired relay based systems, installation time is more as compared to the PLC based control panels.
3. **Availability of Large contacts:** PLC programming tools contain internal large number of contacts that can be used for any change induced in different applications.
4. **Cost effective:** Advanced technology and large production of PLC makes it cheaper than the other controller or relay based systems.
5. **Simulation feature:** PLC programming software comes with the simulation features by default.
6. **Simple programming methods:** PLC is provided with simple programming methods to program the PLC like Ladder or Boolean type of programming.
7. **Ease of maintenance:** As compared with the control systems like relay based or micro-controller based systems, maintenance cost of PLC is low.
8. **Documentation:** The programmer can program and print easily the programs of PLC for future use.

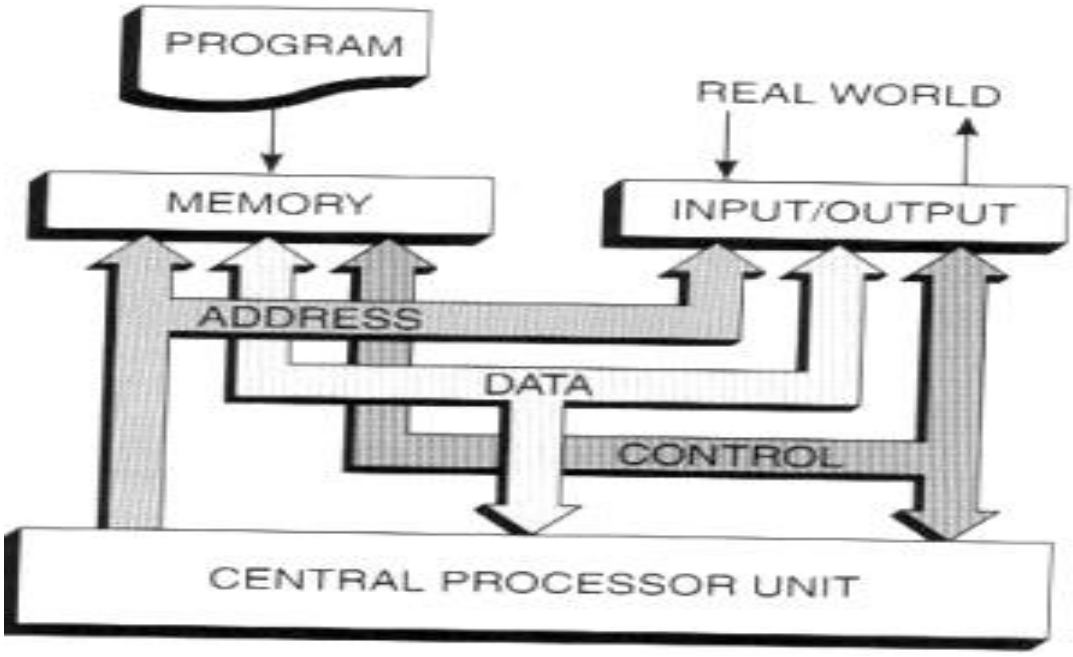
Long type questions [7 marks each]

Q-1) **With neat diagram describe the basic operation of PLC. [S-17(Q6-c)], [S-19(Q6-c)]**

Ans:-Principle of Programmable Logic Controller:

Programmable Logic Controllers are used for continuously monitoring the input values from sensors and produces the outputs for the operation of actuators based on the program. Every PLC system comprises these three modules:

- CPU module
- Power supply module
- One or more I/O module



CPU Module:

A CPU module consists of central processor and its memory. The processor is responsible for performing all the necessary computations and processing of data by accepting the inputs and producing the appropriate outputs.

Power Supply Module:

This module supplies the required power to the whole system by converting the available AC power to DC power required for the CPU and I/O modules. The 5V DC output drives the computer circuitry.

I/O Modules:

The input and out modules of the programmable logic controller are used to connect the sensors and actuators to the system to sense the various parameters such as temperature, pressure and flow, etc. These I/O modules are of two types: digital or analog.

Communication Interface Modules:

These are intelligent I/O modules which transfers the information between a CPU and communication network. These communication modules are used for communicating with other PLC's and computers, which are placed at remote place or far-off locate.

The program in the CPU of programmable logic controller consists of operating system and user programs. The purpose of the operating system with CPU is to deal with the tasks and operations of the PLC such as starting and stopping operations, storage area and communication management, etc. A user program is used by the user for finishing and controlling the tasks in automation.

Q-2) **Describe briefly the concept of PLC programming with example. [S-19(Q7-c)]**

Ans:-PLC programs are typically written in a special application on a personal computer, then downloaded by a direct-connection cable or over a network to the PLC. The program is stored in the PLC either in battery-backed-up RAM or some other non-volatile flash memory. Often, a single PLC can be programmed to replace thousands of relays. The most

commonly used programming language is Ladder diagram (LD) also known as Ladder logic. It uses Contact-Coil logic to make programs like an electrical control diagram.

Control example shown in ladder diagram:-

This is a programming example in ladder diagram which shows the control system. A ladder diagram is a method of drawing control circuits which pre-dates PLCs. The ladder diagram resembles the schematic diagram of a system built with electromechanical relays.

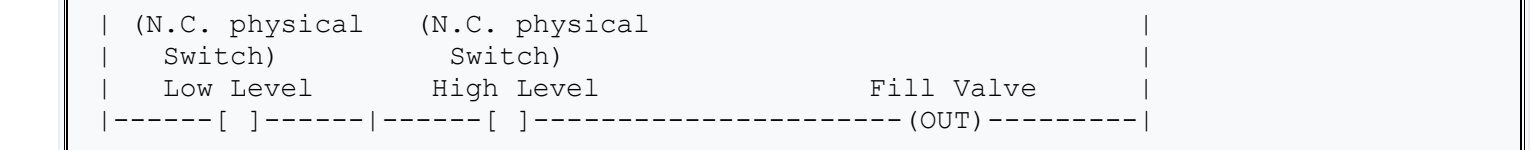
As an example, say a facility needs to store water in a tank. The water is drawn from the tank by another system, as needed, and our example system must manage the water level in the tank by controlling the valve that refills the tank. Shown are:

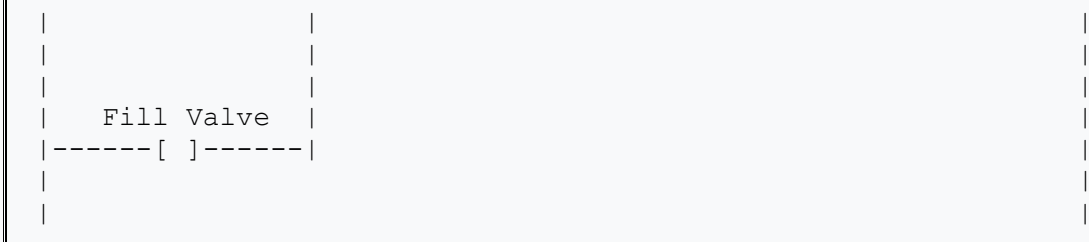
- Two inputs (from the low and high level switches) represented by contacts of the float switches
- An output to the fill valve, labelled as the fill valve which it controls
- An "internal" contact, representing the output signal to the fill valve which is created in the program.
- A logical control scheme created by the interconnection of these items in software

In ladder diagram, the contact symbols represent the state of bits in processor memory, which corresponds to the state of physical inputs to the system. If a discrete input is energized, the memory bit is a 1, and a "normally open" contact controlled by that bit will pass a logic "true" signal on to the next element of the ladder. Therefore, the contacts in the PLC program that "read" or look at the physical switch contacts in this case must be "opposite" or open in order to return a TRUE for the closed physical switches. Internal status bits, corresponding to the state of discrete outputs, are also available to the program.

In the example, the physical state of the float switch contacts must be considered when choosing "normally open" or "normally closed" symbols in the ladder diagram. The PLC has two discrete inputs from float switches (Low Level and High Level). Both float switches (normally closed) open their contacts when the water level in the tank is above the physical location of the switch.

When the water level is below both switches, the float switch physical contacts are both closed, and a true (logic 1) value is passed to the Fill Valve output. Water begins to fill the tank. The internal "Fill Valve" contact latches the circuit so that even when the "Low Level" contact opens (as the water passes the lower switch), the fill valve remains on. Since the High Level is also normally closed, water continues to flow as the water level remains between the two switch levels. Once the water level rises enough so that the "High Level" switch is off (opened), the PLC will shut the inlet to stop the water from overflowing; this is an example of seal-in (latching) logic. The output is sealed in until a high level condition breaks the circuit. After that the fill valve remains off until the level drops so low that the Low Level switch is activated, and the process repeats again.




Fill Valve