10. Explain about I/O programming.

» I/O port programming is an essential aspect of working with the 8051 microcontroller.

» The 8051 microcontroller has four ports:

Port 0 (P0)
Port 1 (P1)
Port 2 (P2)
Port 3 (P3)

» These ports can be used for both input and output operations, allowing you to interface with external devices such as sensors, displays, and other peripherals.

Common Operations for I/O port programming:

1) Configuring I/O Ports:

» Before using I/O ports you need to configure it as either input or output.

» Each port pin can be individually configured as input (logic high impendance) or output (logic low impendance) using corresponding bits in port's associated register.

Ex: To configure P1 as output and P2 as input.

```
MOV P1, #0xFF
MOV P2, #0x00
```

ii) **Reading from Input Ports:**

» To read status of input pins of a port, you can use port's associated register.

» The values read from register indicate logic level of pins.

» A logic high(1) indicates pin is receiving high voltage, while a logic low(0) indicates low voltage.

Ex: To read status of P3 and store in accumulator(A).

```
MOV A,P3
```

iii) **Writing to output Ports:**

» To see the output values of pins in an output port, you can use port's associated register.

» Writing a logic high(1) to pin sets it to a high voltage level and logic low(0) indicates it is set to low voltage level.

Ex: To set P0 to high and P1 to low.

```
MOV P0,#0xFF
MOV P1,#0x00
```

iv) **Manipulating Individual Pins:**

» Sometimes, we need to manipulate individual pins of ports, while keeping others unchanged.

» We can achieve this using bit-wise operations such as AND, OR, XOR, and bit shifting.

Ex: To toggle state pin of P2.3 while keeping other pins of P2 unchanged.

```
CPL P2.3
```

Q) External Pull-Up/Down Resistors:

» The 8051 microcontroller doesn't have internal Pull-Up/Down resistors.

» If you want to use external pull-up/down resistors for input pins, you need to connect them externally to the corresponding pins.

Q) Write about addressing modes of 8051.

» Addressing modes is defined as way of specifying operand in an instruction.

» In 8051 there are 6 types of addressing modes:

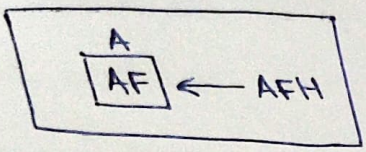
i) Immediate addressing mode
ii) Register addressing mode
iii) Direct addressing mode
iv) Register Indirect addressing mode
v) Indexed Addressing mode
vi) Implied addressing mode

i) Immediate addressing mode:

» In this, data is provided in instruction itself. The data is provided immediately after opcode. Data constant value.

Example:

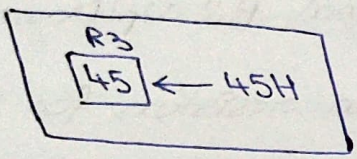
MOV A, #0AFH



→ Immediate data

DPTR → data pointer points external data memory location.

MOV R3, #45H



MOV DPTR, #FE00H

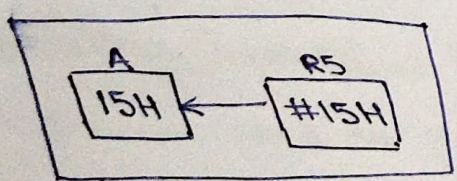
ii) Register addressing mode:

» In this, source or destination data should be present in register (R0 to R7).

Example:

1) MOV A, R5;
MOV R2, #45H
MOV R0, A

2) MOV A, R5
MOV R5, #15H



Register to Register transfer.

iii) Direct addressing modes:

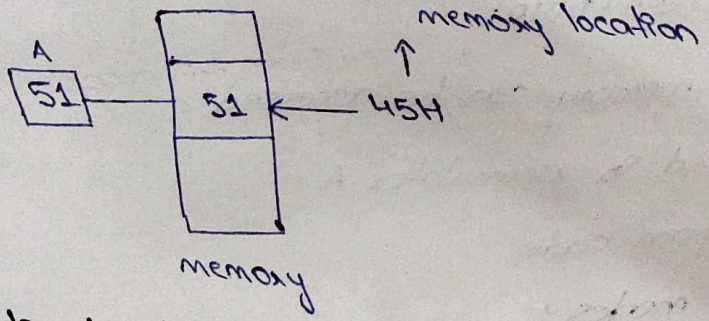
» In this, source or destination data is specified by 8-bit data instruction.

» Only internal memory can be used in this mode.

Example:

1) MOV 80H, R6
   MOV R2, 45H
   MOV R0, 05H

2) MOV R0, 45H
   ADD A, 51H



iv) Register indirect addressing modes:

» In this, source or destination address is given in register.

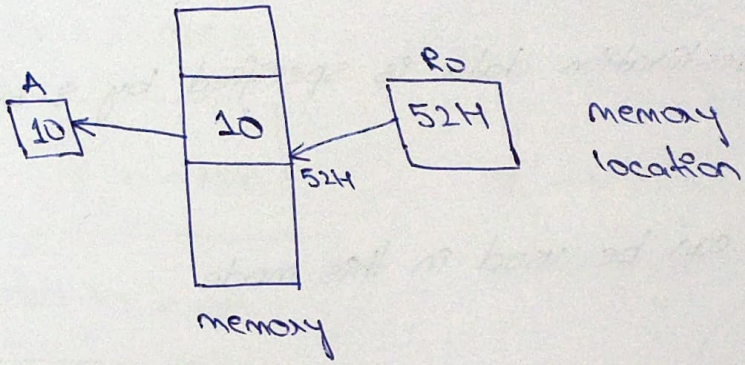
» By using this, internal or external addresses can be addressed.

» The R0 and R1 are used for 8-bit addresses and DPTR is used for 16-bit addresses, no other registers can be used for addressing purposes.

Example:

MOV A, @R0

@ → points address



v) Indexed addressing modes:

» In this, source memory can only be accessed by program only.

» The destination operand is always register A.

Example: MOV CA,@A+PC;

MOV CA,@A+DPTR;

NOTE: Only program memory can be accessed

» Destination operand is accumulator A.

» C refers to program code.

vi) Implied addressing modes:

» In this, there will be single operand.

» These types of instruction can work on specific registers only.

» These type of instructions are also known as register specific instruction.

Example:

RLA;

SWAPA;