

Section - 2

Question :- Enlist and explain the interrupt plus in 8085. Explain the addressing modes of 8085.

Answer :- Interrupts in 8085

Interrupts are the signals generated by the external devices to request the microprocessor to perform a task. There are 5 interrupt signals i.e. TRAP, RST 7.5, RST 6.5, RST 5.5 and INTR.

Interrupts are classified into following groups based on their parameters:-

- Vector interrupt - In this type of interrupt, the interrupt address is known to the processor. For example RST 7.5, RST 6.5, RST 5.5, TRAP.

- Non-Vector Interrupts - In this type of interrupt, the interrupt address is not known to the processor so, the interrupt address needs to be sent externally by the device to produce interrupts. For example INTR.
- Maskable Interrupts - In this type of interrupt we can disable the interrupt by writing some instructions in the program.
- Non-Maskable Interrupts - In this type we can't disable the interrupt by writing some instructions in the program.
- Software Interrupts - In this type of interrupt, the programmer has to add the instructions into the program to execute the interrupt. There are 8 software interrupts in 8085 i.e. RST0, RST1, RST2, RST3, RST4, RST5, RST6 and RST7.

Addressing modes in 8085

These are the instructions used to transfer the data from one register to another register, from the memory to the register, and from the register to the memory without any alteration in the content. Addressing modes in 8085 is classified into 5 groups.

- Immediate addressing mode:-
In this mode the 8/16-bit data is specified in the instruction itself as one of its operand.
- Register addressing mode:-
In this mode, the data is copied from one register to another.

• Direct addressing mode

In this mode, the data is directly copied from the given address to the register.

• Indirect addressing mode

In this mode, the data is transferred from one register to another by using the address pointed by the register.

• Implied register mode

In this mode, the data is transferred from one register to another by using the address pointed by the register.