

Practical No. 15

Title: Implementation of Simple Programs Using I/O-Based Interface

Objective: To develop simple programs that use **Input/Output (I/O) operations** for data transfer between the microprocessor and external devices.

Theory:

An **I/O-based interface** enables communication between the microprocessor and external peripherals such as **keyboards, displays, sensors, and memory devices**. In microprocessors like **8085**, the I/O ports are accessed using **IN** and **OUT** instructions.

I/O Instructions in 8085 Microprocessor:

- **IN port address** → Transfers data from an input port to the accumulator (A register).
- **OUT port address** → Sends data from the accumulator to an output port.

Types of I/O Interfacing:

1. **Memory-Mapped I/O:** Uses memory addresses for I/O devices.
2. **Port-Mapped I/O:** Uses dedicated I/O addresses.

Simple I/O Programs:

1. Reading Data from an Input Port (IN Instruction)

Objective: Read an 8-bit value from an input device (e.g., a switch) and store it in the accumulator.

Program (8085 Assembly Language):

LXI H, 8000H ; Load memory location 8000H

IN 30H ; Read data from input port 30H

MOV M, A ; Store the input data at memory location 8000H

HLT ; Halt execution

Explanation:

- Reads an 8-bit input from port **30H**.
- Stores the data in memory location **8000H**.

2. Sending Data to an Output Port (OUT Instruction)

Objective: Send an 8-bit value to an output device (e.g., an LED display).

Program (8085 Assembly Language):

MVI A, 55H ; Load accumulator with data (55H)

OUT 40H ; Send data to output port 40H

HLT ; Halt execution

Explanation:

- Loads the accumulator with **55H**.
 - Sends the data to the output port **40H** (e.g., LED display).
-

3. Read Data from an Input Port and Display it on an Output Port

Objective: Read an 8-bit value from an input port and send it to an output port.

Program (8085 Assembly Language):

IN 30H ; Read data from input port 30H

OUT 40H ; Send data to output port 40H

HLT ; Halt execution

Explanation:

- Reads an input from port **30H** (e.g., a switch).
 - Displays the same data on output port **40H** (e.g., LED display).
-

Materials/Tools Required:

- 8085 Microprocessor Kit
 - Assembler/Simulator
 - Input device (Switch, Keyboard)
 - Output device (LED, Seven-Segment Display, LCD)
 - Computer with programming software
-

Procedure:

1. **Load the program** into the microprocessor using an assembler/simulator.
 2. **Connect the input/output devices** to the I/O ports.
 3. **Execute the program** to read input and display the result on an output device.
 4. **Observe the data transfer** between the microprocessor and peripherals.
-

Observations:

- The microprocessor successfully **reads input** from an I/O device and **displays output** on an external device.
- The **IN and OUT instructions** control data transfer effectively.
- Using different **port addresses**, multiple input and output devices can be controlled.

Conclusion:

Simple I/O-based programs were successfully implemented using **8085 microprocessor instructions**. These programs enable microprocessors to **interact with external peripherals**, forming the foundation of **embedded systems and real-time processing**.

Applications (Optional):

- **Microcontroller-based automation** (e.g., controlling lights and sensors).
- **Data acquisition systems** (e.g., reading sensors and displaying values).
- **Embedded systems** (e.g., digital clocks, traffic signal controllers).