Practical No. 14

Title: Introduction to 8086 Microprocessor Kit

Objective: To study and understand the **architecture, components, and working** of the **8086 Microprocessor Kit**, which is used for programming and hardware interfacing.

Theory:

Introduction to the 8086 Microprocessor:

The **8086 microprocessor** is a **16-bit microprocessor** developed by **Intel**. It follows the **Von Neumann architecture** and operates in **minimum and maximum mode**. The **8086 supports pipelining**, allowing **faster execution** of instructions. It operates at **5 MHz to 10 MHz clock speed** and has a **20-bit address bus**, enabling it to address **1MB of memory**.

Features of the 8086 Microprocessor:

- **16-bit data bus** (processes 16-bit data at a time).
- **20-bit address bus** (can access **1MB memory**).
- **6 general-purpose registers (AX, BX, CX, DX, SP, BP, SI, DI)**.
- **Pipelining (fetch-execute-overlap process for faster execution)**.
- **Two operating modes**:
    - **Minimum Mode** (Single processor system).
    - **Maximum Mode** (Multiprocessor system).
- **Segmented Memory Architecture**: The memory is divided into four segments:
    - **Code Segment (CS)** – Stores executable instructions.
    - **Data Segment (DS)** – Stores variables and data.
    - **Stack Segment (SS)** – Stores stack operations.
    - **Extra Segment (ES)** – Used for additional data storage.

Components of the 8086 Microprocessor Kit:

The **8086 Microprocessor Kit** is a development board that helps in executing assembly language programs. It contains:

1. **8086 CPU** – The processing unit of the kit.
2. **Memory Unit (ROM & RAM)** – Stores programs and data.
3. **Clock Generator (8284 IC)** – Generates clock pulses for synchronization.
4. **I/O Ports (8255 IC)** – Used for interfacing external devices like LEDs and keyboards.
5. **Interrupt Controller (8259 IC)** – Handles hardware interrupts.

6. **Bus Controller (8288 IC)** – Controls system bus operations.
7. **Power Supply (+5V DC & +12V DC)** – Provides power to the kit.
8. **Monitor Program (Stored in ROM)** – Helps in program execution.

Materials/Tools Required:

- **8086 Microprocessor Kit**
- **Power Supply (+5V and +12V DC Adapter)**
- **LEDs, Switches, and 7-Segment Display (for interfacing)**
- **Assembler Software (for programming)**
- **Computer with Serial Interface (if required for programming)**

Procedure:

1. **Power ON the Kit:**
    - Connect the **+5V and +12V power supply** to the microprocessor kit.
    - Ensure that the system is **properly initialized**.
2. **Understanding the Keyboard and Display:**
    - If the kit has a **hexadecimal keyboard**, use it to enter machine code.
    - Use the **7-segment display or LCD screen** (if available) to view results.
3. **Writing and Executing a Simple Program:**
    - Write a simple assembly language program (e.g., adding two numbers).
    - Enter the machine code into the kit memory.
    - Execute the program using the **RUN command**.
    - Observe the output on the display.
4. **Interfacing with External Devices:**
    - Connect **LEDs, switches, or sensors** to the I/O ports.
    - Write a program to **control LEDs or read switch inputs**.
5. **Reset and Shutdown:**
    - Use the **RESET button** to restart the kit.
    - Power OFF the kit after execution.

Observations:

- The **8086 microprocessor kit** successfully runs **assembly language programs**.
- Input and output devices can be controlled using **IN and OUT instructions**.
- The memory unit stores programs and processes data efficiently.
- **Pipelining improves processing speed** compared to the 8085 microprocessor.

Conclusion:

The 8086 Microprocessor Kit provides a **practical platform** to understand **low-level programming, hardware interfacing, and microprocessor operations**. It plays a crucial role in **embedded systems, robotics, and industrial automation**.

Applications :

- Used in **embedded systems development**.
- Applied in **traffic signal control, automation, and industrial control systems**.
- Essential for **teaching microprocessor programming and interfacing techniques**.