|  |
| --- |
| Experiment No. 8 |
| Implement 8086 assembly language program for performing sorting of array |
| Date of Performance: |
| Date of Correction: |

**Aim:** To implement an 8086-assembly language program that sorts an array of numbers in ascending or descending order.

**Objective:** To understand and apply sorting logic at the assembly language level using 8086 microprocessor instructions.

**Theory:**

**Introduction:**

Sorting is a basic yet vital operation in computer science used to organize data for faster searching, better readability, and optimal storage. In high-level languages, we often use built-in functions for sorting, but at the **assembly language** level, the programmer must implement the logic explicitly using low-level instructions and register operations.

In this practical, we will write an assembly program using **8086 microprocessor instructions** to sort an array of numbers. The commonly used sorting technique in assembly is **Bubble Sort** due to its simple logic and suitability for step-by-step implementation using loops and comparisons.

**8086 Microprocessor Recap:**

The Intel 8086 is a **16-bit processor** with a rich instruction set for:

* **Data transfer**
* **Arithmetic and logical operations**
* **Looping and branching**
* **Memory addressing and pointer manipulation**

In sorting, we use instructions such as:

* MOV – move data
* CMP – compare operands
* JAE, JB, JE, etc. – conditional jumps
* XCHG – swap values
* LOOP – loop control
* Registers: SI, DI, CX, AX, BX, etc.

**Sorting Algorithm (Bubble Sort) Logic:**

**Bubble Sort** works by repeatedly comparing adjacent elements and swapping them if they are in the wrong order.

Steps:

1. Load the array into memory.
2. Use a loop to traverse the array.
3. Compare each pair of adjacent elements.
4. Swap them if they are in the wrong order.
5. Repeat this process n-1 times (for n elements).

**Example (Ascending Order):**  
Input Array: [6, 3, 9, 1]  
Pass 1: [3, 6, 1, 9]  
Pass 2: [3, 1, 6, 9]  
Pass 3: [1, 3, 6, 9]

Bubble sort is simple and easy to implement in assembly due to its reliance on **pairwise comparisons** and **adjacent swaps**.

**Real-World Applications:**

* Understanding how high-level sorting functions work at machine level.
* Developing firmware for devices with limited computing resources.
* Learning how **memory and registers interact** in assembly language.
* Enhancing skills in **pointer arithmetic**, **conditional jumps**, and **loop structures**.
* Essential for **embedded systems**, **OS-level programming**, and **microcontroller firmware**.

**Solution:**

**Conclusion:** We implemented a sorting algorithm in 8086 assembly language, which helped us understand the use of loops, comparisons, and memory access at the processor level.