

Practical: 10

Aim: Write an Assembly Language Program for finding smallest number from an array in 8086.

Theory:

org 100h: This directive indicates that the program is a .COM file. It tells the assembler where to start placing the code in memory.

array db ...: This defines an array of 10 bytes with the values given. This is the data we will be working with.

smallest db 0: This variable will store the smallest number found in the array.

arraySize db 10: This defines the size of the array, which is 10.

.code: This indicates the beginning of the code segment.

start: This is a label marking the beginning of the main code execution.

mov al, array[0]: Load the first element of the array (34) into the AL register.

mov smallest, al: Initialize the smallest variable with the first element of the array.

- **mov cl, arraySize:** Load the size of the array (10) into the CL register (part of the CX register).
- **mov si, 1:** Initialize the index SI to 1, as we have already considered the first element.

Finding the Smallest Number

- **find_smallest:** This label marks the beginning of the loop to find the smallest number.
- **cmp si, cl:** Compare the current index (SI) with the size of the array (CL).
- **je done:** If they are equal, it means we have checked all elements, so jump to the done label.

Loading and Comparing Elements

- **mov al, array[si]:** Load the current array element into AL.
- **cmp al, smallest:** Compare the current element with the smallest found so far.
- **jge next:** If the current element is greater than or equal to smallest, jump to the next label.

Updating the Smallest Number

- **mov smallest, al:** If the current element is smaller, update smallest with the new value.

Move to Next Element

- **next:** This label marks the point to continue to the next element.
- **inc si:** Increment the index to check the next element.
- **jmp find_smallest:** Jump back to the start of the find_smallest loop.

Done Label

- This label marks the end of the loop when the smallest number has been found.

Output the Smallest Number

- **mov al, smallest:** Load the smallest number into AL.
- **add al, '0':** Convert the number to its ASCII representation by adding the ASCII value of '0'.
- **mov ah, 0Eh:** Set up the BIOS teletype function to print a character.
- **int 10h:** Call the BIOS interrupt to display the character in AL.

Exit the Program

mov ax, 4C00h: Prepare to terminate the program.

int 21h: Call the DOS interrupt to exit the program.

Algorithm to Find the Smallest Number in an Array

1. **Initialize Variables:**
 - Define an array of numbers.
 - Initialize a variable to store the smallest number, starting with the first element of the array.
 - Define a variable for the size of the array.
2. **Set Up Loop:**
 - Set the index variable to start from the second element of the array (index 1).
 - Load the size of the array into a counter.
3. **Loop Through the Array:**
 - **Check the Loop Condition:**
 - Compare the current index with the size of the array.
 - If the index equals the size, exit the loop.
 - **Load Current Element:**
 - Load the current element of the array into a temporary variable.
 - **Compare Current Element:**
 - Compare the current element with the smallest number found so far.
 - If the current element is smaller, update the smallest number.
4. **Increment Index:**
 - Increment the index variable to move to the next element in the array.
5. **Repeat:**
 - Repeat the process until all elements in the array have been checked.
6. **Output the Result:**
 - Convert the smallest number found to its ASCII representation.

- Display the smallest number.

7. Exit Program:

- Terminate the program cleanly.

Conclusion: Hence, we have implemented an Assembly Language Program for finding smallest number from an array in 8086.