

Implementation of Bubble Sort Using 8086 Assembly language

Md. Tanbin Hossain Himel
2018-1-60-151
Department of CSE, East West
University, Dhaka, Bangladesh
2018-1-60-151@std.ewubd.edu

Nasim Bahadur
2018-1-60-073
Department of CSE, East West
University, Dhaka, Bangladesh
2018-1-60-073@std.ewubd.edu

Golam Kibria Shaqur
2018-1-60-237
Department of CSE, East West
University, Dhaka, Bangladesh
2018-1-60-237@std.ewubd.edu

Abstract— Bubble sort is one of the most renowned algorithms in computer science. In this project our goal is to implement bubble sort using 8086 assembly language. Also, we learnt how the assembly language work.

Keywords— Assembly language 8086, Bubble sort.

I. INTRODUCTION The assembly programming language is a low-level language which is developed by using mnemonics. The microcontroller or microprocessor can understand only the binary language like 0's or 1's therefore the assembler converts the assembly language to binary language and store it the memory to perform the tasks. Before writing the program, the embedded designers must have sufficient knowledge on hardware of the controller or processor, so first we required to know hardware of 8086 processor.

The 8086 is a processor that is represented for all peripheral devices such as serial bus, and RAM and ROM, I/O devices and so on which are all externally connected to CPU by using a system bus. The 8086 microprocessor has CISC based architecture, and it has peripherals like 32 I/O, Serial communication, memories, and counters/timers. The microprocessor requires a program to perform the operations that require a memory for read and save the functions.

The assembly level programming 8086 is based on the memory registers. A Register is the main part of the microprocessors and controllers which are in the memory that provides a faster way of collecting and storing the data. If we want to manipulate data to a processor or controller by performing multiplication, addition, etc., we cannot do that directly in the memory where need registers to process and to store the data. The 8086 microprocessor contains various kinds of registers that can be classified according to their instructions such as:

A. General purpose registers: The 8086 CPU has consisted of 8-general purpose registers and each register has its own name as shown in the figure such as AX, BX, CX, DX, SI, DI, BP, SP. These all are 16-bit registers where four registers are divided into two parts such as AX, BX, CX, and DX which is mainly used to keep the numbers.

- B. Special purpose registers: The 8086 CPU has consisted of 2- special function registers such as IP and flag registers. The IP register point to the current executing instruction and always works to gather with the CS segment register. The main function of flag registers is to modify the CPU operations after mechanical functions are completed and we cannot access directly
- C. Segment registers: The 8086 CPU has consisted of 4-segment registers such as CS, DS, ES, SS which is mainly used for possible to store any data in the segment registers and we can access a block of memory using segment registers.

Now, our target is to implement bubble sort using 8086 assembly language.

II. METHODOLOGY Our first target was to learn assembly language 8086. After learning it, we memorize bubble sort since we learn it in our algorithm course. Then we write a C++ code. Finally, we convert it to assembly language 8086.

III. SYSTEM & TOOLS REQUIREMENTS To complete this project we need to have followings systems and tools.

- Processor: Intel Core i3 3.40GHz or equivalent
- RAM: 8GB
- Operating system: Windows 10
- EMU8086
- DOSBOX
- NOTEPAD++ etc.

BUBBLE SORT

Algorithm of Bubble Sort:

```
begin BubbleSort(list)
  for all elements of list
    if list[i] > list[i+1]
      swap(list[i], list[i+1])
    end if
  end for
  return list
end BubbleSort
```

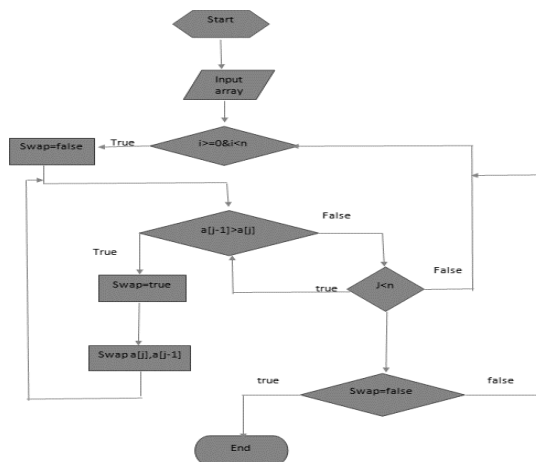
Pseudocode of Bubble Sort:

```

procedure bubbleSort( list : array of items )
loop = list.count;
for i = 0 to loop-1 do:
  swapped = false
  for j = 0 to loop-1 do:
    /* compare the adjacent elements */
    if list[j] > list[j+1] then
      /* swap them */
      swap( list[j], list[j+1] )
      swapped = true
    end if
  end for
  /*if no number was swapped that means
  array is sorted now, break the loop.*/
  if(not swapped) then
    break
  end if
end for
end procedure return list

```

Flow Chart for Bubble Sort:



Steps for Implementing Bubble sort IN 8086 Assembly language:

- Declare an array
- Set all elements to 0
- Take n inputs in the array
- Start a loop of n iteration
- Compare index i to index i+1
- Swap element if index i is greater
- Iterate the loop n times
- Print the sorted array

IV. RESULT ANALYSIS Here is the output of our code:

```

emulator screen (80x25 chars)
Enter list(10 values without space):9803126410
Unsorted list: 9 8 0 3 1 2 6 4 1 0
Sorted list : 0 0 1 1 2 3 4 6 8 9

```

V. LIMITATIONS During this project we face some challenges:

- Cannot input Negative values
- Cannot input multidigit number
- Need exactly n values to run the program
- The number must be fit in the 16-bit register

VI. FUTURE WORK In our project we are unable to handle negative values & multi digit numbers. Also, our array size is fixed. Therefore, there is some scope to work with this in the future.

VII. CONCLUSION In this project, we implement bubble sort using 8086 assembly language. As a conclusion, we can say that 8086 language is too much complex than higher level language. As well as we learn the importance of having higher level programming language which makes our life easier.

REFERENCES

- [1] "Contents (exploded view) - IBM Documentation." <https://www.ibm.com/docs/en/zos/2.1.0?topic=hla-sm-language-reference> (accessed Sep. 11, 2021).
- [2] J. A. Saxon, *Programming the IBM 1401, a self-instructional programmed manual*. Englewood Cliffs, N.J., 1962.
- [3] "Assembly: Review Lecture 14."
- [4] "Assembly Language For Students | Guide books." <https://dl.acm.org/doi/book/10.5555/3125846> (accessed Sep. 11, 2021).