Simulator Manual
Advance Computing Engine

CS222

Computer Organization and Assembly Language

Submitted By:

Ahad Lodhi – 2020042

Mohsin Zia – 2020244

Syed Irtaza – 2020474

Zartaj Asim – 2020526

Submitted To:

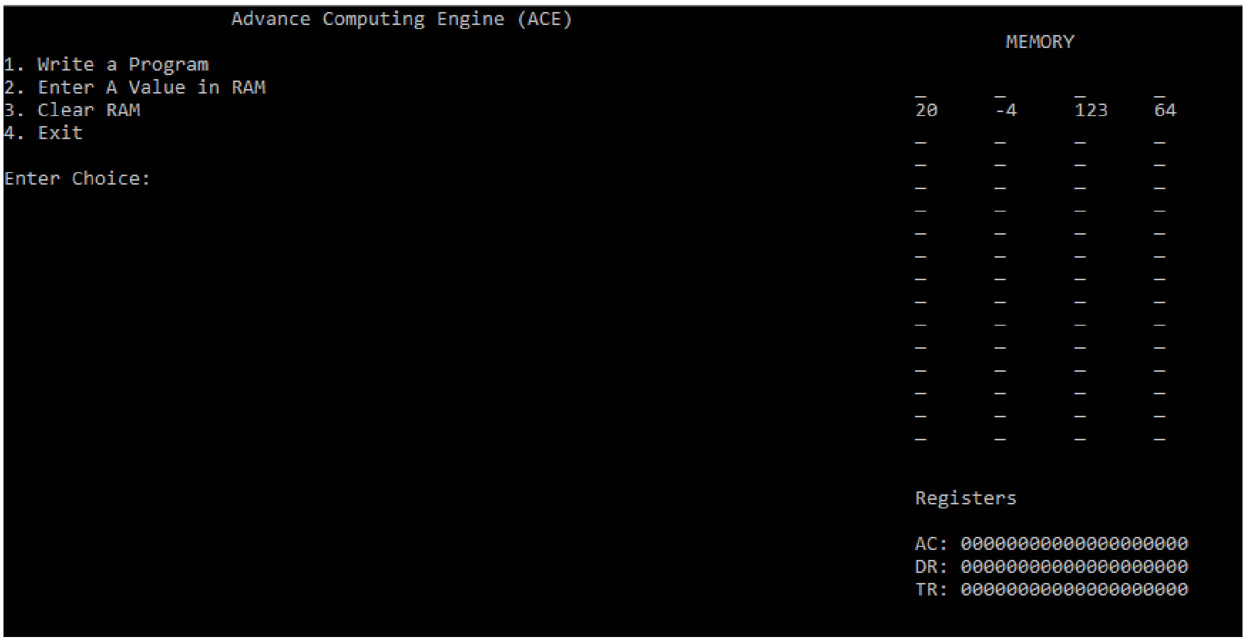
Dr. Ghulam Abbas

Date: 25-May-2022

Introduction:

Advance Computing Engine (ACE) is the computer that is made by our group as a project of this course. For better understanding of this project, a C++ simulator is also created. This simulator shows most of the function that ACE can do but however this simulator is not the full practical version of this. This simulator is for idea of how ACE works. Memory Unit in this simulator contains (0-63) 64 words. Also, three registers: Accumulator, Data Register, and Temporary Register are included in this. This simulator is a menu-driven simulator which gives user an ease to operate this and update the Memory Unit and Registers after every instruction being executed. Also, it takes instruction one by one and executes that.

Working:



```
                    Advance Computing Engine (ACE)
                                                                                            MEMORY
1. Write a Program
2. Enter A Value in RAM                                                              _      _      _      _
3. Clear RAM                                                                         20     -4     123    64
4. Exit                                                                              _      _      _      _
                                                                                     _      _      _      _
Enter Choice:                                                                        _      _      _      _
                                                                                     _      _      _      _
                                                                                     _      _      _      _
                                                                                     _      _      _      _
                                                                                     _      _      _      _
                                                                                     _      _      _      _
                                                                                     _      _      _      _
                                                                                     _      _      _      _
                                                                                     _      _      _      _
                                                                                     _      _      _      _
                                                                                     _      _      _      _
                                                                                     _      _      _      _

                                                                                     Registers

                                                                                     AC: 00000000000000000000
                                                                                     DR: 00000000000000000000
                                                                                     TR: 00000000000000000000
```

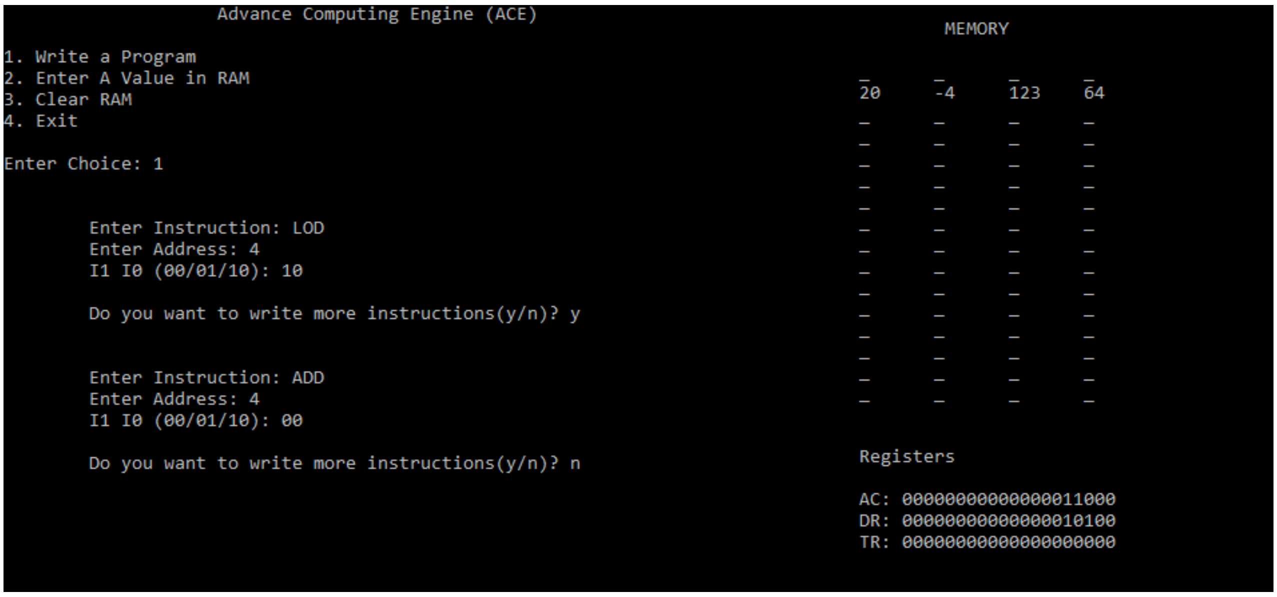
Initially, Memory has some values in 4-7 words. User can make choice to: Write a Program, Enter a value in Memory (RAM), Clear Ram, Exit. These all four choices are handled by if else conditions.

1. **Write a Program:**
   This choice let user to write instructions one by one. First user has to enter an instruction (i.e., LOD, ADD, OR, etc.…), after this user has to enter the address, and after the address user has to enter addressing mode (00 for direct, 01 for indirect, 10 for immediate). After user is done entering instructions, Memory and Registers will be updated.

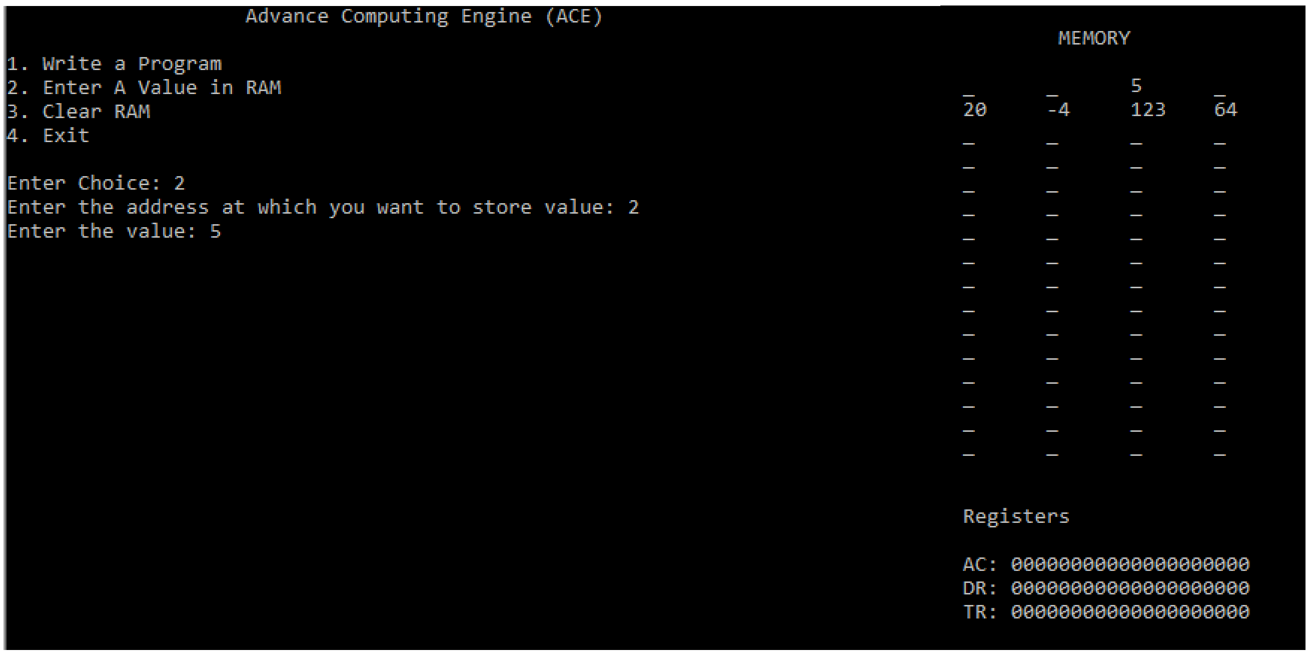
2. **Enter a Value in RAM:**
   It will ask a user to enter address at which the value should be stored. After the address, user should enter a value to be stored.

3. **Clear RAM:**

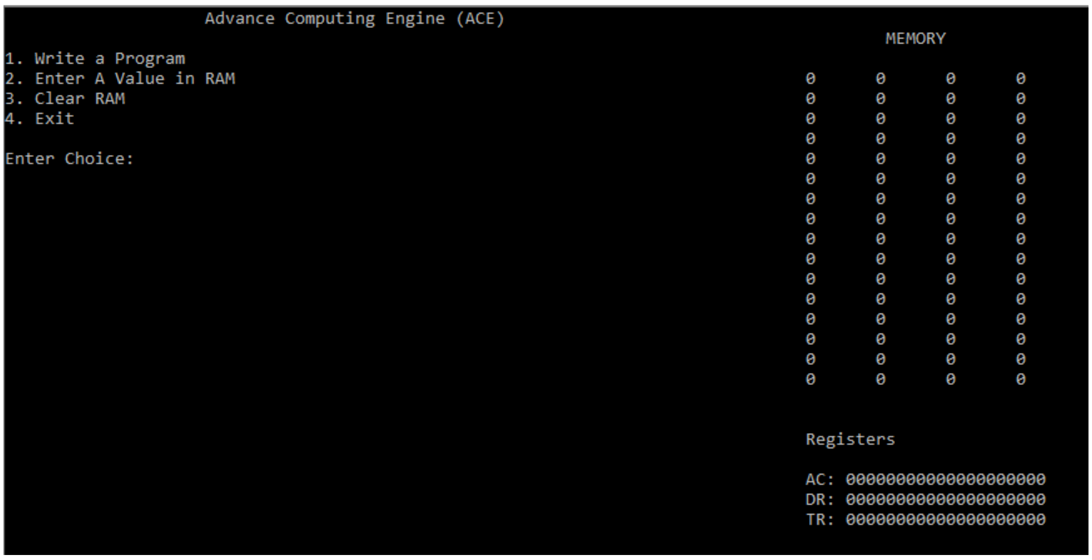
   Initially, RAM has some garbage value (to be exact -999 as hardcoded). To clear the RAM means, store every memory word with 0. Choosing this option will clear the RAM.



```
                    Advance Computing Engine (ACE)
                                                                        MEMORY
1. Write a Program
2. Enter A Value in RAM                                         _      _      _      _
3. Clear RAM                                                    20     -4     123    64
4. Exit                                                         _      _      _      _
                                                                _      _      _      _
Enter Choice: 1                                                 _      _      _      _
                                                                _      _      _      _
                                                                _      _      _      _
        Enter Instruction: LOD                                  _      _      _      _
        Enter Address: 4                                        _      _      _      _
        I1 I0 (00/01/10): 10                                    _      _      _      _
                                                                _      _      _      _
        Do you want to write more instructions(y/n)? y          _      _      _      _
                                                                _      _      _      _
                                                                _      _      _      _
        Enter Instruction: ADD                                  _      _      _      _
        Enter Address: 4                                        _      _      _      _
        I1 I0 (00/01/10): 00

        Do you want to write more instructions(y/n)? n          Registers

                                                                AC: 00000000000000011000
                                                                DR: 00000000000000010100
                                                                TR: 00000000000000000000
```



```
                    Advance Computing Engine (ACE)
                                                                        MEMORY
1. Write a Program
2. Enter A Value in RAM                                         _      _      5      _
3. Clear RAM                                                    20     -4     123    64
4. Exit                                                         _      _      _      _
                                                                _      _      _      _
Enter Choice: 2                                                 _      _      _      _
Enter the address at which you want to store value: 2           _      _      _      _
Enter the value: 5                                              _      _      _      _
                                                                _      _      _      _
                                                                _      _      _      _
                                                                _      _      _      _
                                                                _      _      _      _
                                                                _      _      _      _
                                                                _      _      _      _
                                                                _      _      _      _
                                                                _      _      _      _
                                                                _      _      _      _

                                                                Registers

                                                                AC: 00000000000000000000
                                                                DR: 00000000000000000000
                                                                TR: 00000000000000000000
```



```
                    Advance Computing Engine (ACE)
                                                                        MEMORY
1. Write a Program
2. Enter A Value in RAM                                         0      0      0      0
3. Clear RAM                                                    0      0      0      0
4. Exit                                                         0      0      0      0
                                                                0      0      0      0
Enter Choice:                                                   0      0      0      0
                                                                0      0      0      0
                                                                0      0      0      0
                                                                0      0      0      0
                                                                0      0      0      0
                                                                0      0      0      0
                                                                0      0      0      0
                                                                0      0      0      0
                                                                0      0      0      0
                                                                0      0      0      0
                                                                0      0      0      0
                                                                0      0      0      0

                                                                Registers

                                                                AC: 00000000000000000000
                                                                DR: 00000000000000000000
                                                                TR: 00000000000000000000
```

Explanation:

Each node represents an instruction, and linked list is used to take instructions from user, so there is no bound-on number of instructions. Then in insert function, user enter instruction. If instructions are memory referenced then user is asked to enter an address and addressing mode, else if instruction is register referenced then no address is required. Each instruction is checked by if-else and executed respectively.

The memory instructions require the user to enter two things:

1. Enter the address
2. Enter the address mode

00 for direct addressing

01 for indirect addressing

10 for immediate addressing

The instructions used in our simulator are as follow:

1. LOD

The load instruction is used to load the data in accumulator.

2. ADD

It adds the memory contents into the data in accumulator.

3. MUL

The Mul instruction is used to multiply the content into the data in accumulator.

4. DIF

The Difference instruction is used to subtract two contents into the accumulator.

5. INC

The increment instruction is used to increment the content in the accumulator and store them.

6. DEC

The decrement instruction is used to decrement the content in the accumulator and store it.

7. SWP

The swap value is used to swap two values in the accumulator.

8. NOT

The not instructions nots the content in the accumulator.

9. OR

The or instruction is used to or the values in the accumulator.

10. NOR

The Nor instruction nor the values in the accumulator.

11. AND

The and instruction is used to and the contents in the accumulator.

12. NAND

The nand instruction nand’s the contents in the accumulator.

13. XOR

The Xor instruction xor’s the contents in the accumulator.

14. XNOR

The Xnor instruction Xnor’s the contents in the accumulator.

15. GTR

If memory content is greater than accumulator then it updates the value of the accumulator.

16. LES

If memory content is less than accumulator then it updates the value of the accumulator.

17. CLA

Clears accumulator.

18. CMA

Compliment the contents in the accumulator.

19. TCA

It performs two’s complement the contents in the accumulator.

20. SHL

Shift the content in the accumulator towards the left.

21. SHR

Shift the content in the accumulator towards the right.