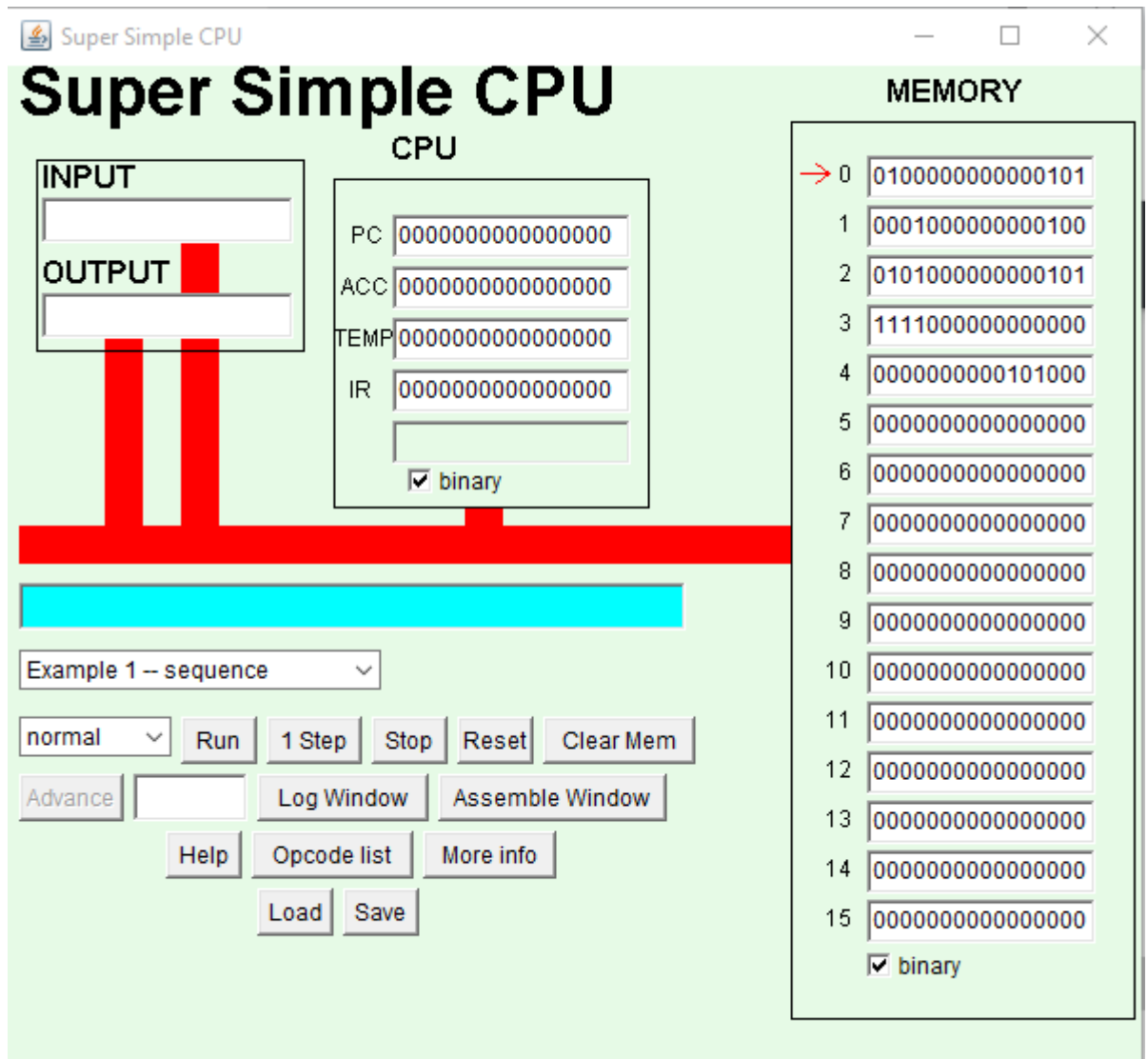# LAB 4

1. Lab 7 Computer Cycling (Lab7_Manual.pdf)

a. Exercise 1

    1. Use Super Simple CPU app.
    2. Select example 1.
    3. Write down instructions or take a screenshot.

    4. Click on opcodes list.
    5. Decode three instructions and write down the corresponding three-letter mneumonic.

  ANS: At memory location**0**: 0100 that means **LDI**

At memory location**1**: 0001 that means **ADD**

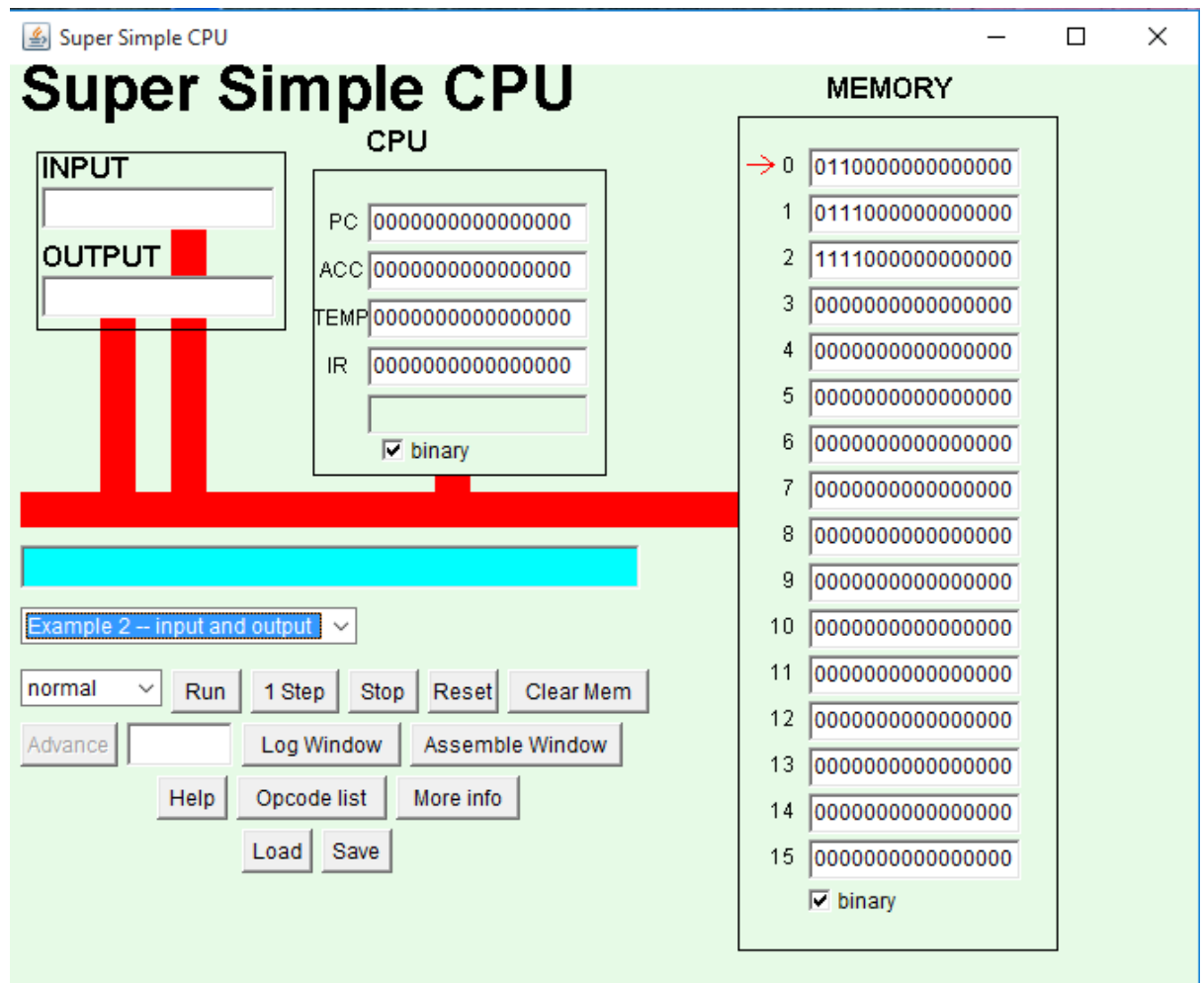At memory location**2**: 0101 that means **STO**

6. How many fetch execute cycles will this perform?

ANS: It performs 4 fetch execute cycles with last(4th) one which stops the computer.

b. Repeat Exercise 1 for examples 2, 3, and 4

1. Example 2

1. Use Super Simple CPU app.
2. Select example 1.
3. Write down instructions or take a screenshot.

4. Click on opcodes list.

5. Decode three instructions and write down the corresponding three-letter mneumonic.

ANS: At memory location**0**: 0110 that means **INP**

At memory location**1:** 0111 that means **OUT**

At memory location**2**: 1111 that means **STP**

6. How many fetch execute cycles will this perform?

ANS: It performs 3 fetch execute cycles with last(3rd) one which stops the computer.

2. Example 3

1. Use Super Simple CPU app.
2. Select example 1.
3. Write down instructions or take a screenshot.

4. Click on opcodes list.
5. Decode three instructions and write down the corresponding three-letter mneumonic.

ANS: At memory location**0**: 0100 that means **LDI**
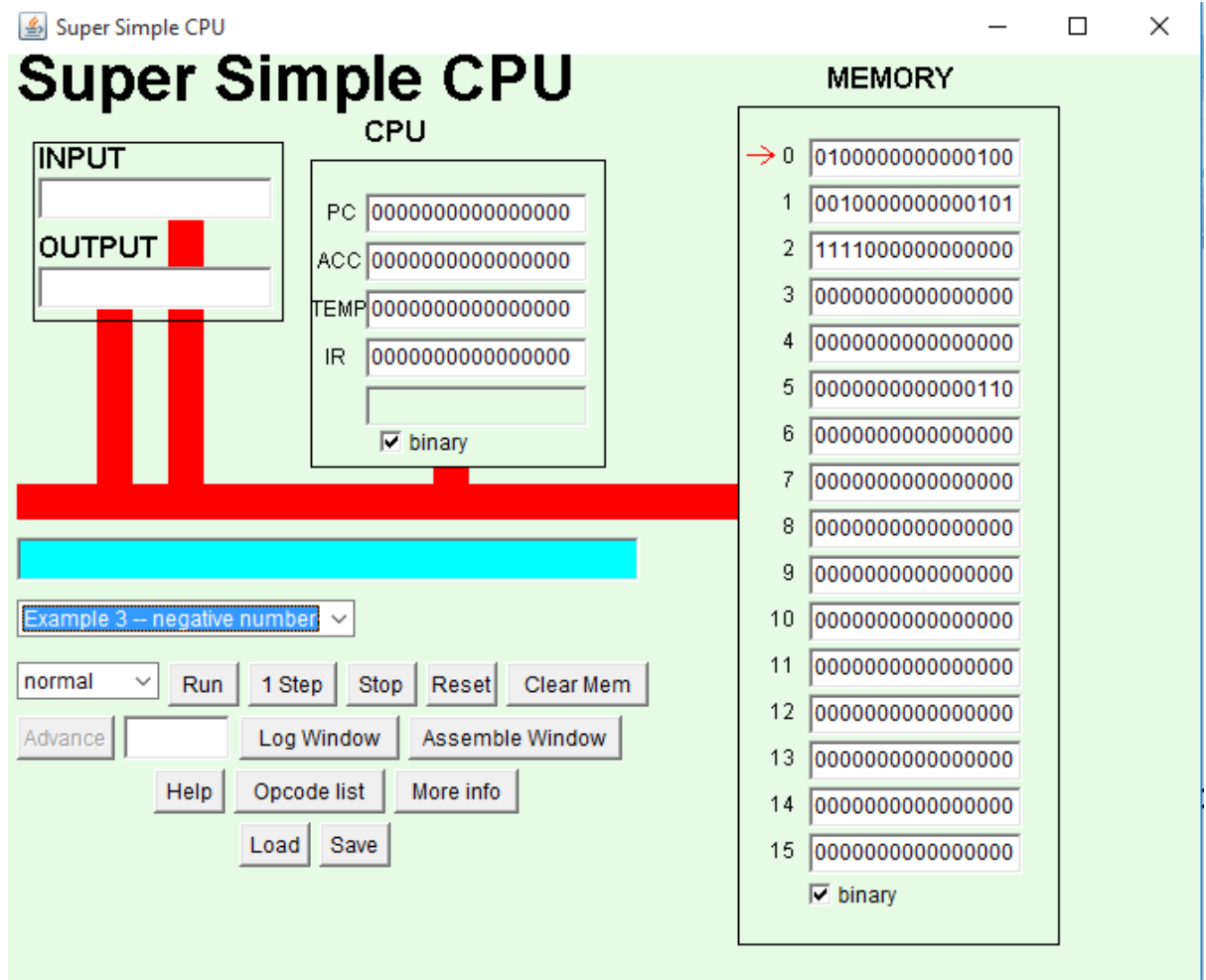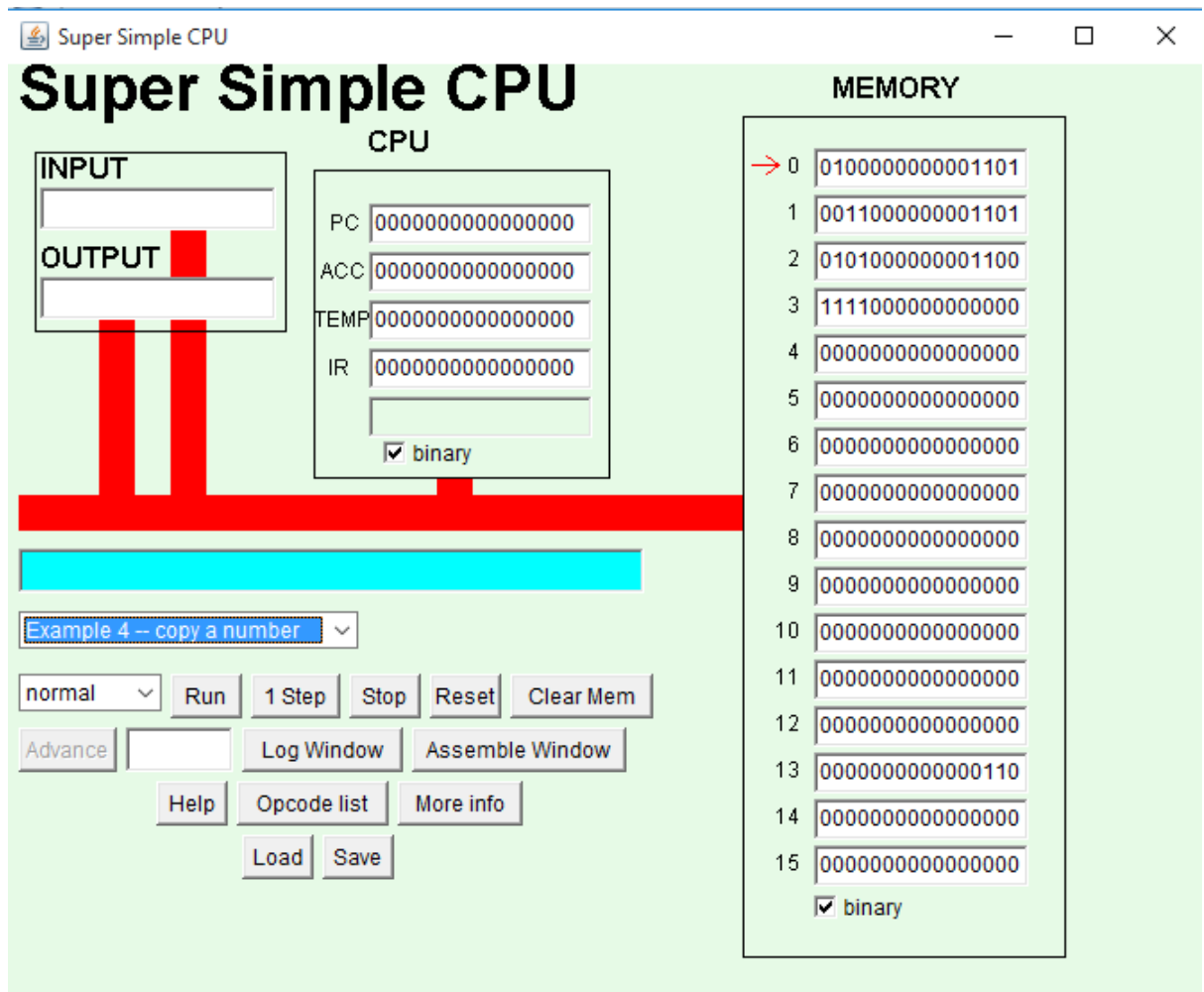
At memory location**1:** 0010 that means **SUB**

At memory location**2**: 1111 that means **STP**

6. How many fetch execute cycles will this perform?

ANS: It performs 3 fetch execute cycles with last(3rd) one which stops the computer.

3. Example 4

1. Use Super Simple CPU app.
2. Select example 1.
3. Write down instructions or take a screenshot.

4. Click on opcodes list.
5. Decode three instructions and write down the corresponding three-letter mneumonic.

ANS: At memory location0: 0100 that means **LDI**

   At memory location1: 0011 that means **LOD**

   At memory location2: 0101 that means **STO**

   At memory location3: 1111 that means **STP**

6. How many fetch execute cycles will this perform?

ANS: It performs 4 fetch execute cycles with last(4th) one which stops the computer.

c. Exercise 2

1. Start the app.
2. Load and run the second example.
3. Now, we'll modify the program: In place of STP instruction in word2, encode a STO (store) instruction that will store the accumulator value in memory cell 15.
4. Create the new STP instruction in word 3, to make sure your program halts.
5. Run your program. When it finishes, you should see your input number in memory cell 15.
6. Take a screenshot.

ANS:
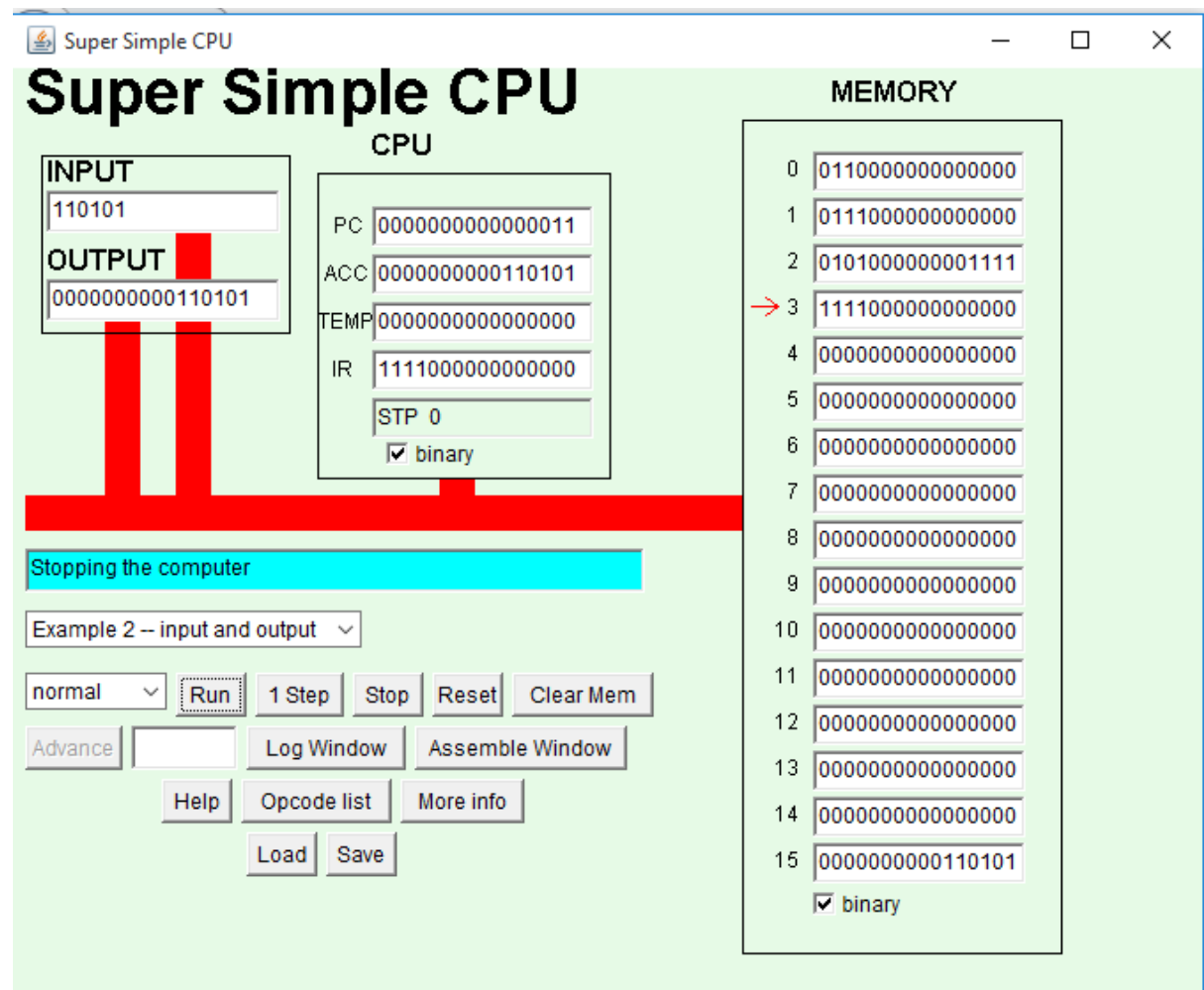
d. Exercise 3

1. Start the Super Simple CPU App.
2. Type instructions in the memory cell 0 and 1.

   0100 0000 0000 1110

   0011 0000 0000 1110

3. In cell 14, type:

   0000 0000 0001 0101

4. Press 1 Step button and write down three letter mnemonic code and value in Accumulator.

ANS: The three letter mnemonic codes are:

For cell 1: 0100   LDI

For cell 2: 0011   LOD

Value in Accumulator is 0000000000001110.

5. Again press 1 Step button and write the mnemonic accumulator's value.
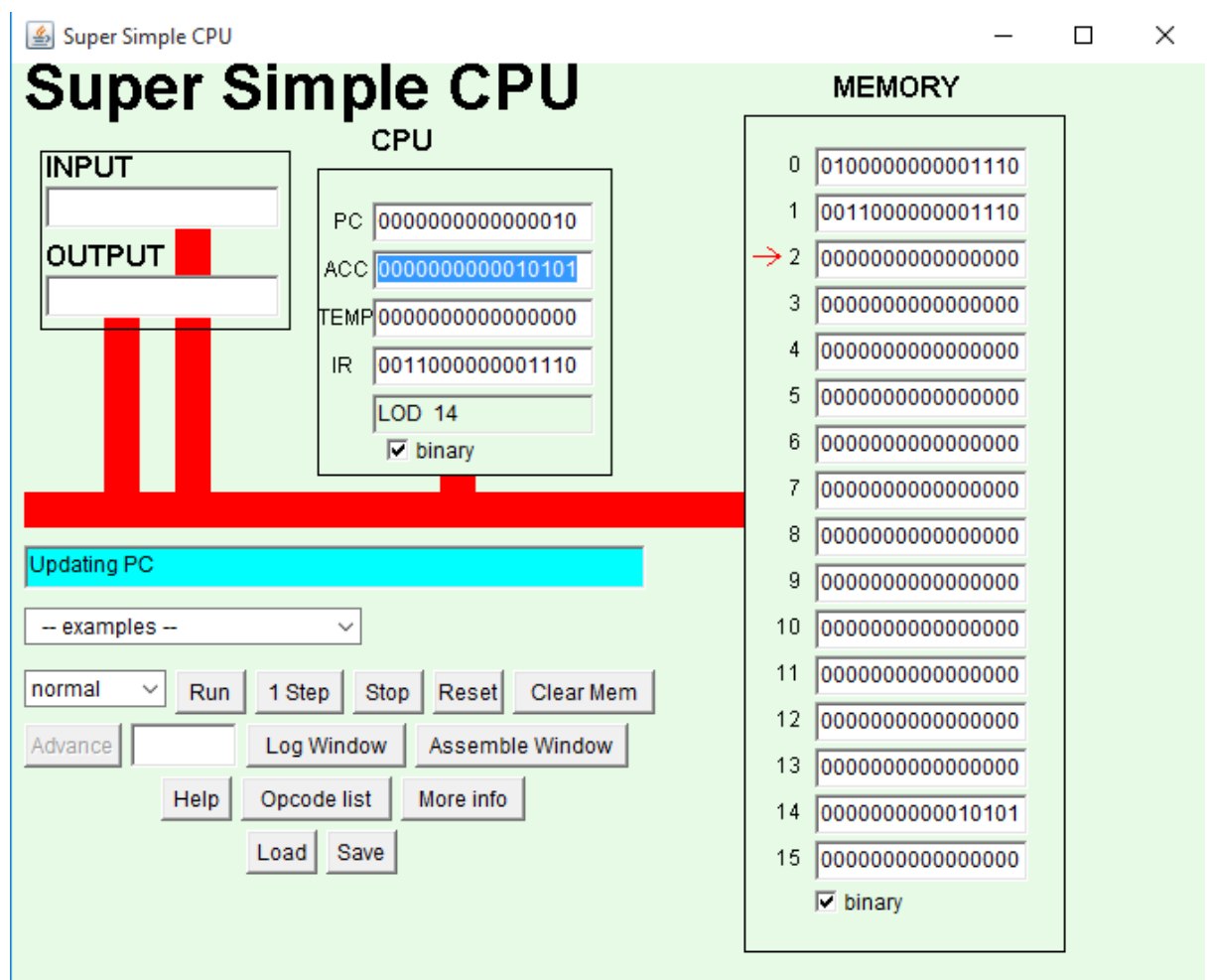
ANS: The three letter mnemonic codes are:

For cell 1: 0100   LDI

For cell 2: 0011   LOD

Value in Accumulator is 0000000000010101.

6. Take a screenshot.

7. Explain the difference between the two instructions.

ANS: The difference between the two instructions is that in first the mnemonic is LDI and the value of the memory cell 0 is copied leaving the mnemonic code while for second step the mnemonic is LOD and in this the value of the memory cell 14 is loaded to Accumulator.

8. Why do you think there is no "store immediate" instruction?

ANS: There is no "store immediate" instruction because there is no STO(Store accumulator in the memory cell) instruction and hence no nothing is stored in the memory cells.

9. What sequence of regular instructions could be used to do the same thing? That is, how could we store a specific number say 133 into a specific memory location, say memory cell 13?

ANS: We can use an alternative to store 133 in memory cell 13 which is:

Step 1: we will load 133 into accumulator using LDI 133

Step 2: we will store the value of accumulator in memory cell 13 using STO 13.

Step 3: Stop the computer using STP

2. What is the printout of the following pseudocode if the user inputs 4 and 10?

Display "Enter two numbers: "

Input num1

Input num2

 Set sum to num1 + num2

 Set average to sum / 2

Display "Average of " + num1 + " and " + num2 + " is " + average

ANS: The print out will be as follows:

Enter two numbers:

4

10

Average of 4 and 10 is 7

3. (Input and Output) Write a pseudocode for example 2 of Super Simple CPU Applet.

4. (Temperature Conversion) Write a pseudocode that:

a. reads a Celsius degree from input

Ans: Print "Enter Celsius degree: " Input Celsius

b. converts it to Fahrenheit (you need the formula to convert from Celsius to Fahrenheit)

Ans: Set Fahrenheit = 1.8(Celsius) +32

c. displays the result as show in the sample run below

Ans: Display "Degree in Fahrenheit is " + Fahrenheit

5. (Simple Calculations) Write a pseudocode that:

a. reads two positive integers

b. calculates sum and product (multiplication) of the numbers

c. displays the result as show in the sample run below.

ANS:

Display "Enter two positive integers: " input num1 input num2

Set sum = num1 + num2

Set product = num1*num2

Display "Sum is " + sum + " . Product is " + product + ".".

6. Given the following "Super Simple CPU" program:

a. Trace the code, write the values of Accumulator, Memory (ONE), and Memory (TWO) in a table (shown below), and record their changes after each instruction.

ANS:

Assembly Instruction	Accumulator	ONE	TWO
LDI 6	6	1	2
SUB ONE	6-1 = 5	1	2
ADD TWO	5+2 = 7	1	2
STO ONE	7	7	2
STP	7	7	2

b. Write a pseudocode that performs the same program.

```
        LDI 6 ;    Load 6 into accumulator

        SUB ONE ;    Subtract the value in memory location ONE

                      from Accumulator

        ADD TWO ;   Add the value in memory location TWO to

                      accumulator

DONE     STO ONE ;   Store accumulator in memory location ONE

        STP ;           Stop the program
```

ONE   DAT 1 ;   A data value, the constant 1

TWO   DAT 2 ; A data value, the constant 2

ANS:

Assembly Instruction	Pseudocode
LDI 6	Set A to 6
SUB ONE	Set A to A-ONE
ADD TWO	Set A to A + TWO
STO ONE	Set ONE to A
STP	

Result: Set ONE to 1

           Set TWO to 2

           Set ONE to 6 – ONE + TWO

7. Given the following "Super Simple CPU" program:

a. Trace the code, write the values of Accumulator, Memory (X), Memory (Y), and Memory (Z) in a table (shown below), and record their changes after each instruction.

ANS:

Assembly Instructions	Accumulator	X	Y	Z
LOD X	3	3	5	0
ADD X	3+3 = 6	3	5	0
ADD X	6 + 3 =9	3	5	0
SUB Y	9-5 = 4	3	5	0
STO Z	4	3	5	4
STP	4	3	5	4

b. Write a pseudocode that performs the same program.

        LOD X ;   Load X into accumulator

        ADD X ;   Add the value in memory location X to

               accumulator

        ADD X ;   Add the value in memory location X to

               accumulator

        SUB Y ;   Subtract the value in memory location Y from

               accumulator

DONE   STO Z ;   Store accumulator in memory location Z

        STP ;   Stop the program

X   DAT 3 ;   A data value, the constant 3

Y   DAT 5 ;   A data value, the constant 5

Z   DAT 0 ;   A data value, the constant 0

ANS:

Assembly Instructions	Pseudocode
LOD X	Set A to X
ADD X	Set A to A+X
ADD X	Set A to A+X
SUB Y	Set A to A-Y
STO Z	Set Z to A
STP	

Final result:

Set X to 3

Set Y to 5

Set Z to 0

Set Z to 3*X - Y