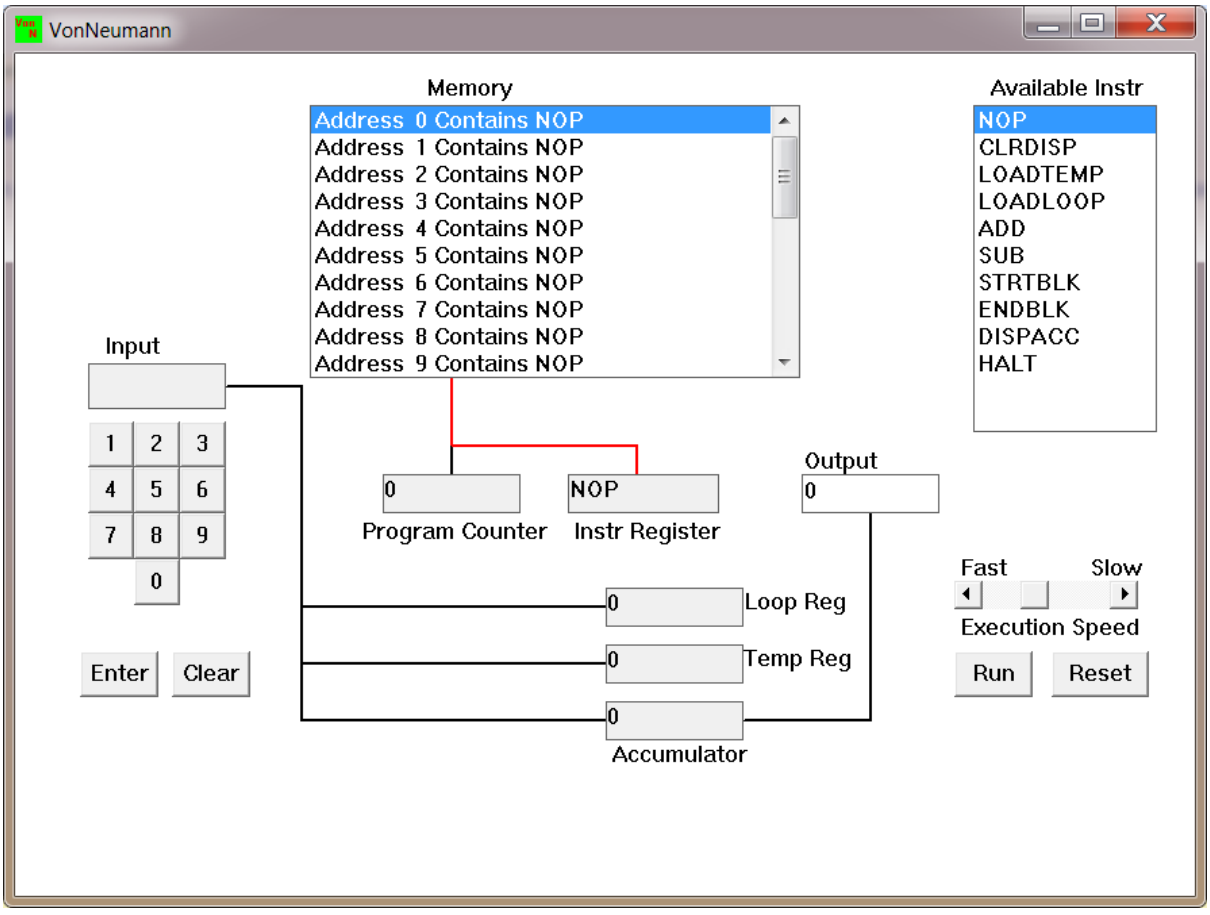
# 4CS015 Fundamentals of Computing – Workshop-6

Name: Kamal Dhital
Student ID: 2407046

**Workshop tasks:**

1. Von Neumann Simulator. This program simulates a very simple computer with the von Neumann architecture.

    a. Download the von Neumann Simulator (VonNeumann.exe) program from WOLF in the week 5 folder. Save it in your Documents folder and run it. You will see a window similar to this:




The simulator has a small program memory area which is available for programming. To enter your program instruction simply click on the “Available” instruction on the list on the right and then click on the “Memory” location you wish to put it in.

This simulator understands only the following ten instructions:

| | |
|---|---|
| NOP | No Operation, i.e. do nothing. |
| LOADTEMP | Get a number from the keypad, completed by the Enter key, into the Temporary Register. |
| LOADLOOP | Get a number from the keypad, completed by the Enter key, into the Loop Register. |
| CLRDISP | Clear the Display. |
| ADD | Add the Temporary Register to the Accumulator |
| SUB | Subtract the Temporary Register from the Accumulator |
| DISPACC | Display the contents of the Accumulator |
| STRTBLK | Start of Loop Block |
| ENDBLK | End of Loop Block |
| HALT | Halt. Stop Program |

b. Load the following program into the memory.

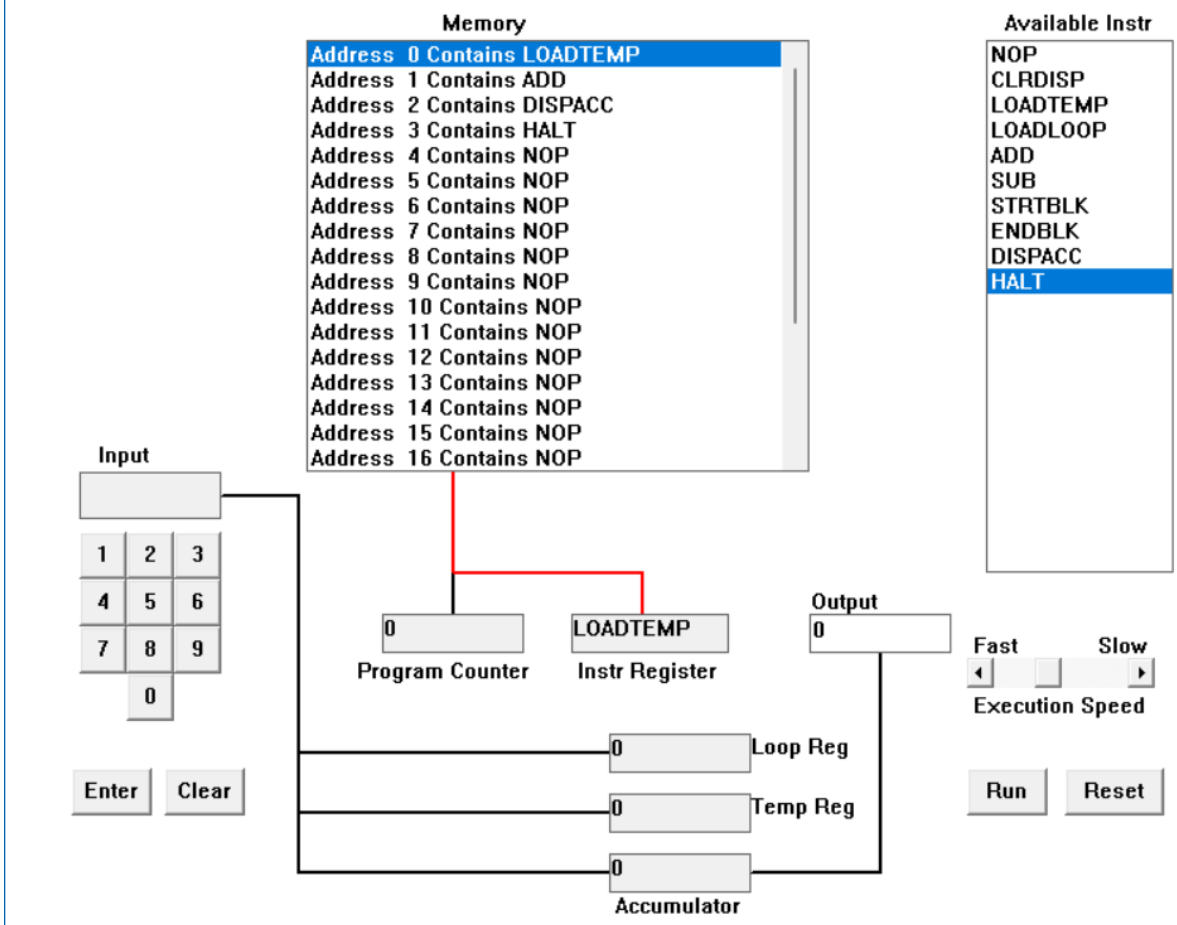
LOADTEMP

ADD

DISPACC

HALT

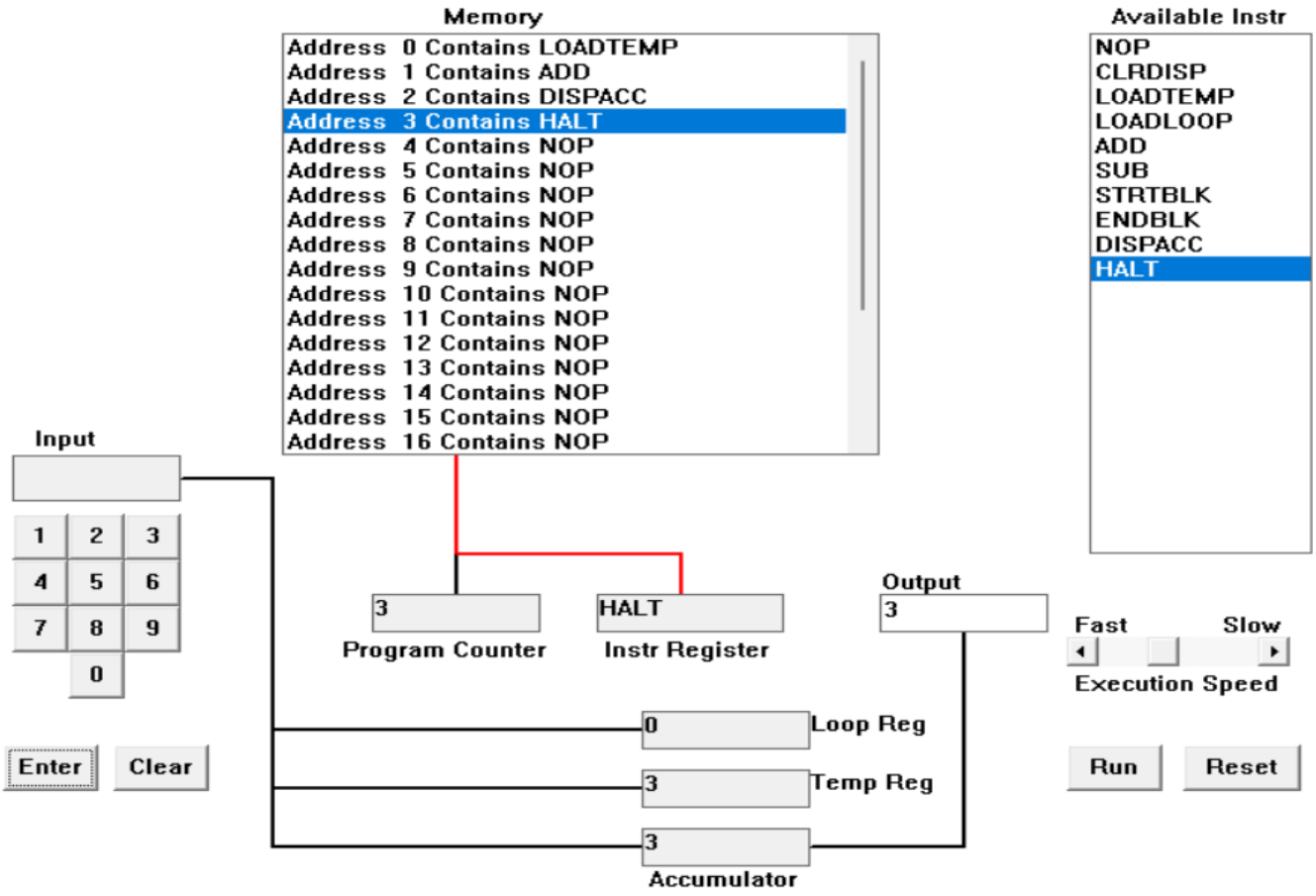
To do this, first click on the “LOADTEMP” in the list of instruction on the right of simulator window. Then click on Memory location with “Address 0 Contains NOP”. This will then change into “Address 0 Contains LOADTEMP”. Repeat the process with “Address 1” and so on until the whole program is loaded.




c. Run the program by clicking on the “Run” button. The simulator would highlight the Address 0 location and then pause. It is executing the instruction “LOADTEMP” which requires you to input a number into the keypad.

Click 2 or 3 numbers on the keypad and then click the “Enter” button. The simulator will then resume running the program and execute the instruction “ADD”. This adds the number that you just entered, to the zero in the accumulator.

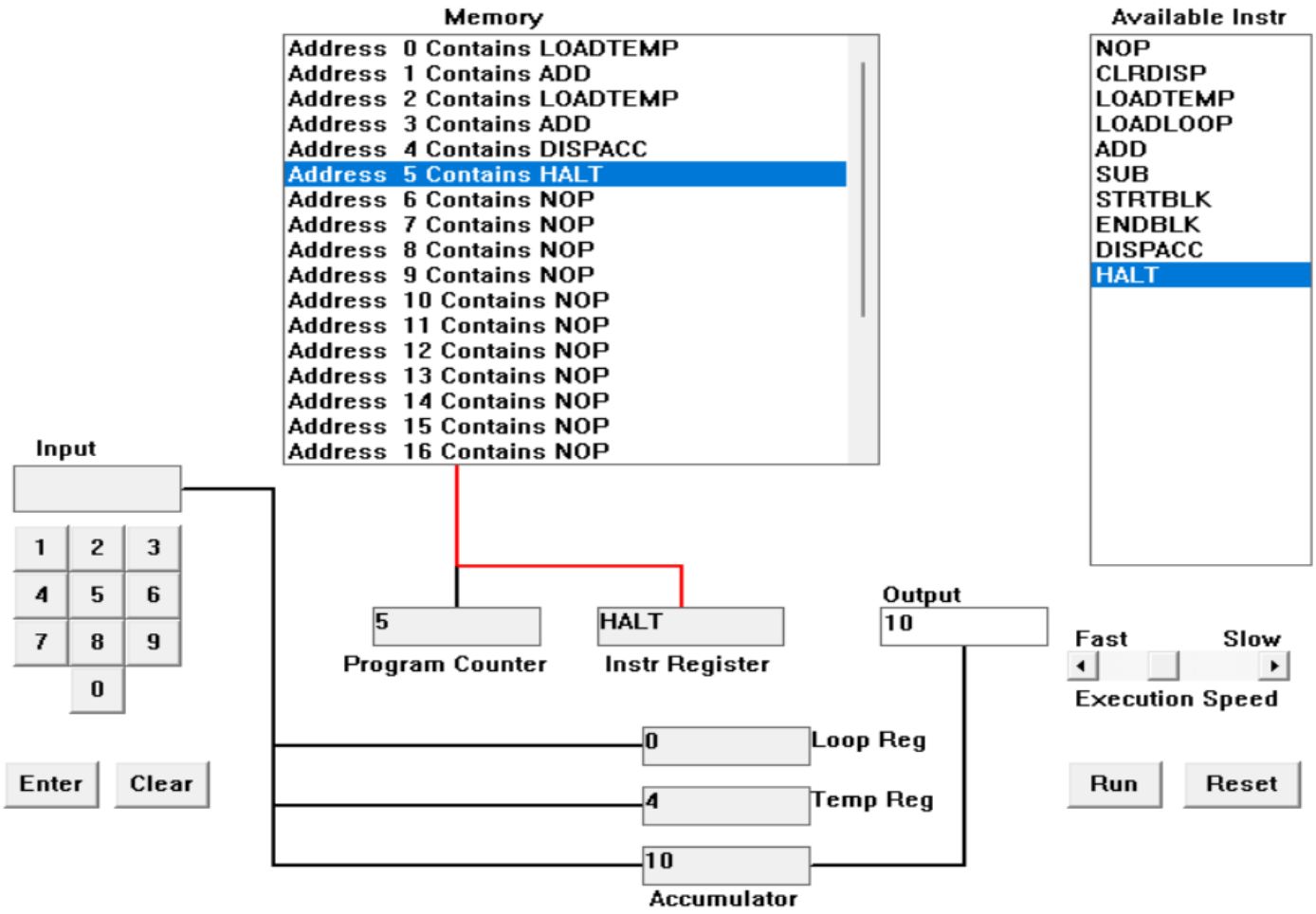
The next instruction is “DISPACC” which stands for “Display Accumulator”, and it does exactly that. After than the simulator stops running the program when it executes the instruction “HALT”.




d. Load the following program into the simulator:

LOADTEMP
ADD
LOADTEMP
ADD
DISPACC
HALT

What do you think it does? Write your anser below (10 marks)




Looking toward the overall figure, we can easily find out what is happening in the program. Memory address gives us the detail about how this program is executed step by step. Program Counter tells us that this program is execute 5 times. Instruction Register show the current position of instruction of instruction in memory address, as an output of it we see HALT which means program is terminate at this position. Loop register have the value 0 which defines no looping is remaining to perform in the program. Temporary register defines the latest values entered by the user. We can see the Accumulator which store the value after each time program runs. At last output is the place where we get the output of our program which is 10 getting from 6 + 4 as and first input and second input.

e. Write a program to add 4 numbers together. List your program below (10 marks)

Code:

LOADLOOP -- 4

STRTBLK

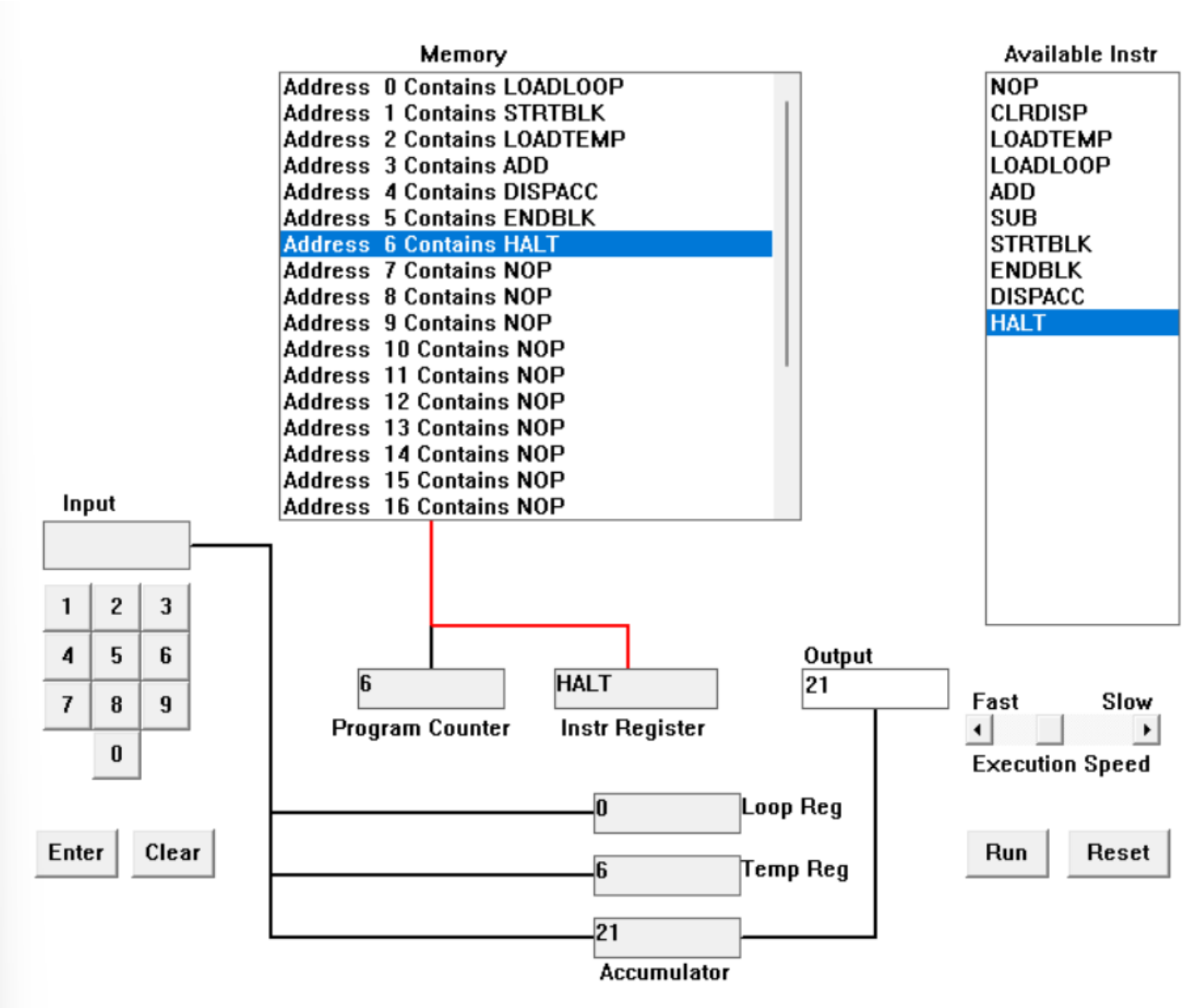
LOADTEMP – 5,3,7,6

ADD

DISPACC

ENDBLK

HALT




In the above figure I run the LOADLOOP to initialize the loop for the four addition in a row, following the loadloop we STRTBLK for it where we put the LOADTEMP to take the number which need to be added and after inserting the LOADTEMP we move forward to the add in next step and add the number provided above and DISPACC is used to store the calculation in accumulator and ENDBLK to closed and start looping from STRTBLK to ENDBLK until the loop remaining becomes 0. Once loop remaining become 0, the HALT in memory address 6 is used to end the program and we can see the

output in accumulator. Program counter defines how many times do this program runs but it counts loop only once. In the above figure, we get the value 21 after adding 4 numbers (5,3,7,6) and we can see last number is 6 in the temp register.

f. Write a program to subtract a number from another. List your program below (10 marks)

LOADTEMP -- 6

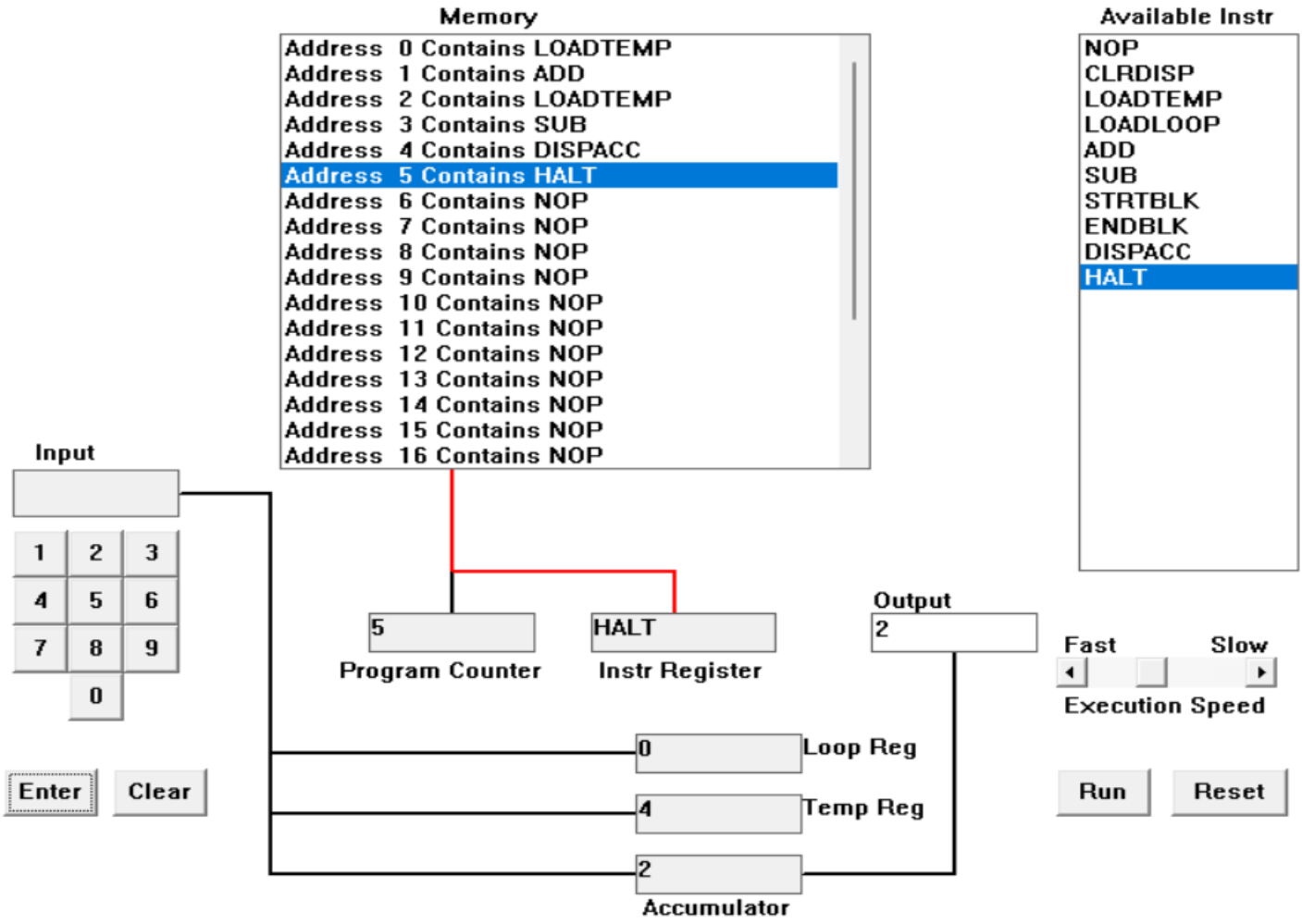
Add

LOADTEMP -- 4

SUB

DISPACC

HALT



The above program subtracts the first number enter by user to the second number entered by the user. This program contains the sequence of LOADTEMP, ADD, LOADTEMP, SUB, DISPACC, HALT. Load temp is used to enter the first number and add is used to perform the addition of 0 and LOADTEMP number where 0 is the temporary value stored in temp register at the beginning. After it, another LOADTEMP is used to load another number and SUB is used to perform the subtraction of first number to second number and DISPACC store the value in the accumulator from where we get the value after program is execute which is HALT in this case.

g. Load the following program into the simulator:

   LOADTEMP – 5

   ADD

   LOADLOOP -- 6

   STRTBLK

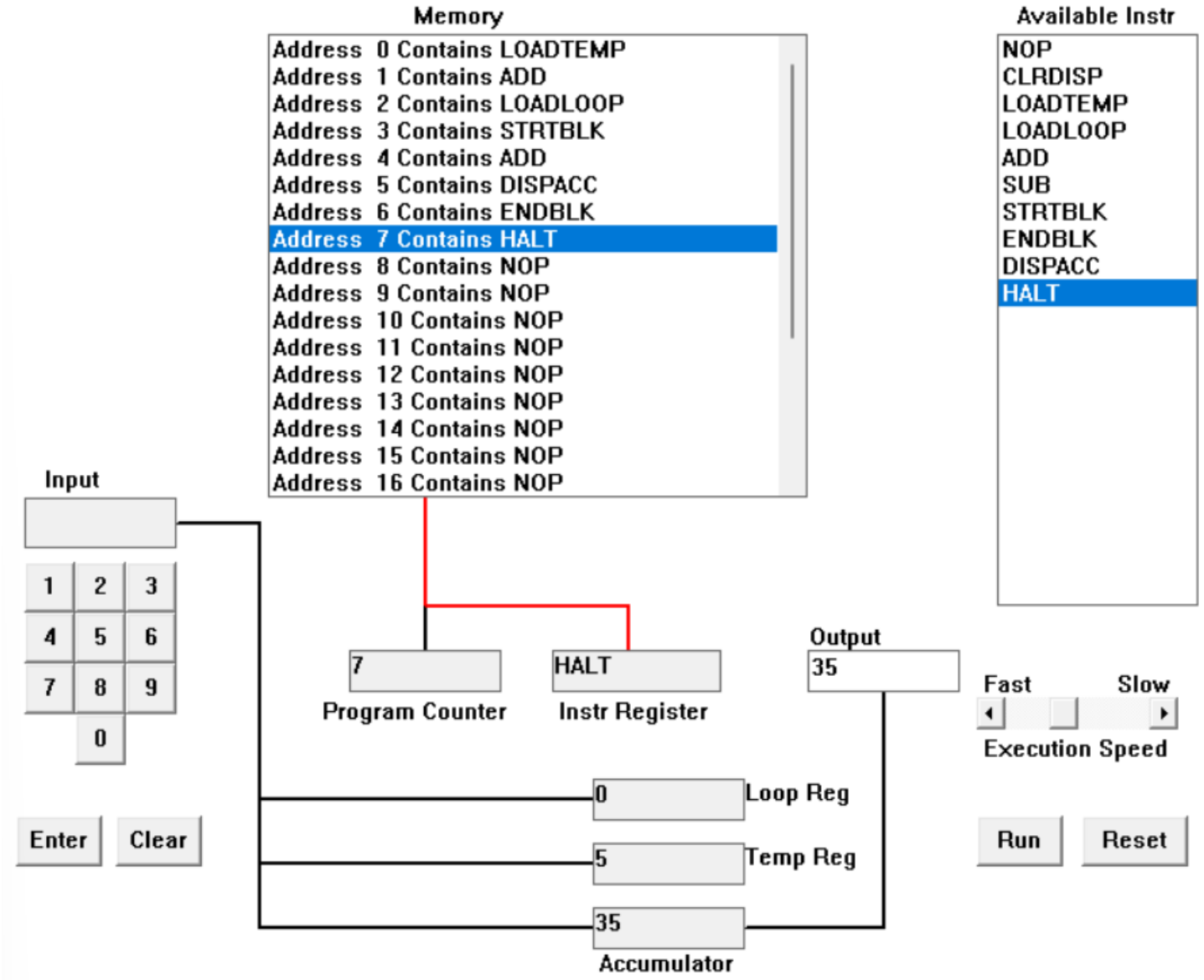
   ADD

   DISPACC

   ENDBLK

   HALT

   Run it and when it reaches the LOADTEMP instruction, enter 5 on the keypad and click the “Enter” button. When it reaches the LOADLOOP instruction, enter 6. What do you think the program does? Write answer below in the form of and equation (10 marks)




Similarly, to other question, let visualize how this code works. LOADTEMP is used to load the first value which we need to given as the number for it this number is added and TEMP REG hold the value of it. After that the LOADLOOP is used to take the value for it for looping. STRTBLK and ENDBLK is used to define the looping condition of above loop. Between STRTBLK and ENDBLK we have two operations to do which are ADD and DISPACC. Inside it the looping run until the Loop Reg become 0 and each time Add is used to add the previous number and DISPACC is used to store the number. When Loop Reg become 0 the looping condition will be stop and it return the value in accumulator and HALT will terminate the program and display the output of it. In this case as question defined the LOADTEMP should be 5 and LOADLOOP will be 6 and looping the 5 six times with adding gives 30 as we do basic calculation. But in this program after we ADD the LOADTEMP number the number in temporary register will be 5 and looping 5 another 6 times gives the result of 35.

h. Write a program that will let you add 5, or 10 or 20 numbers together. List your program below and explain how it works (25 marks)

Code:

LOADLOOP

STRTBLK

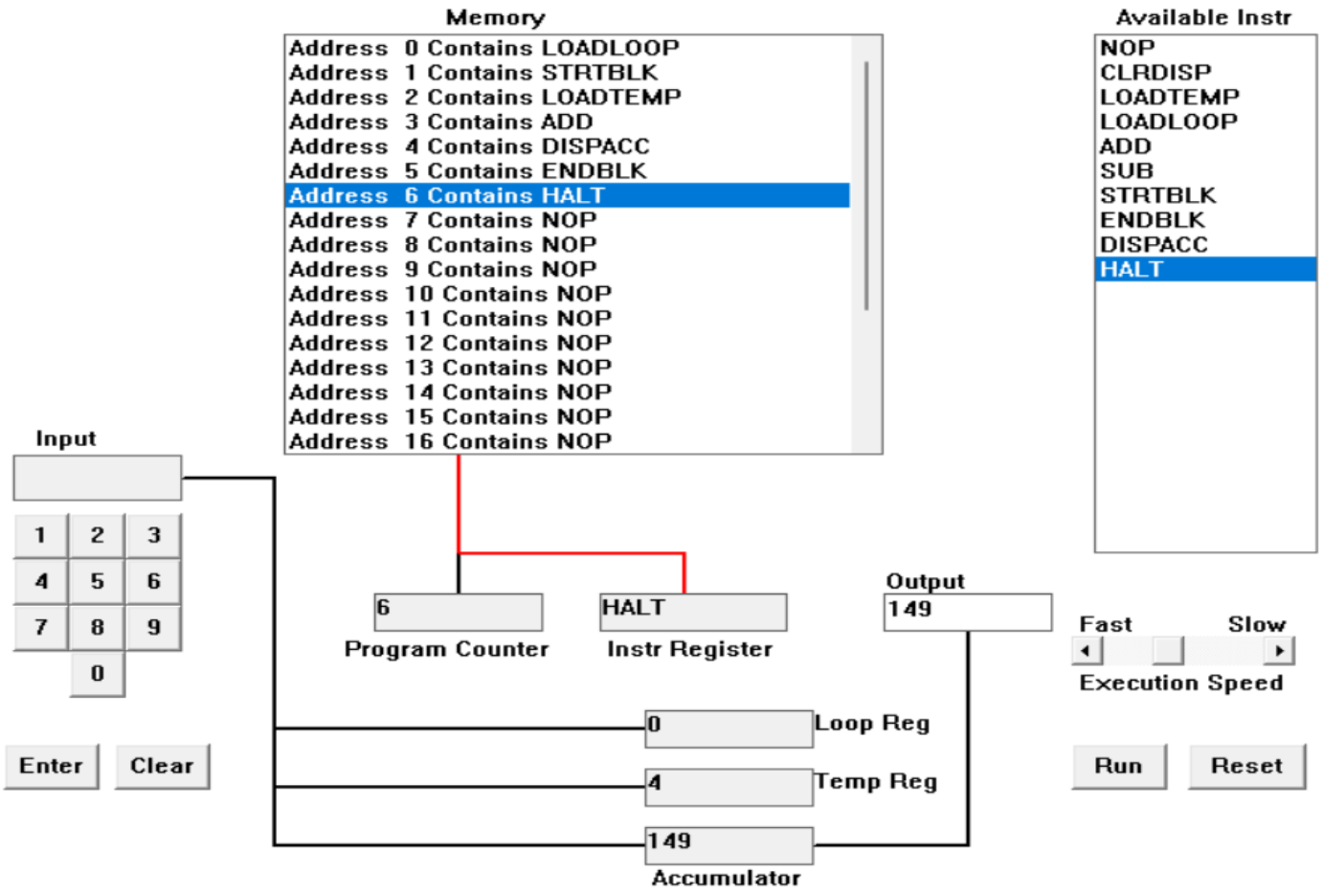
LOADTEMP

ADD

DISPACC

ENDBLK

HALT



This program is used to add the numbers of numbers according to the loop defined by the user. This program contains LOADLOOP, STRTBLK, LOADTEMP, ADD, DISPACC, ENDBLK and HALT. LOADLOOP and LOADTEMP will request the number from the used to define how many times loop run and what numbers are added.

Taking about the component of the above figure, Available instruction is used to give the available instruction to the memory address as per need. The memory address holds the program and execute once we press the Run button. Reset button is used to reset the program to prevent from initial values stored in the program. Program Counter defines how many times our program executes but it only counts loops for 1. Instruction Register defines what instruction is currently running in the program. Loop Reg define the remaining loop to execute in the program. Temp reg stand for temporary register which register the latest value. Input is used to enter the number for LOADLOOP and LOADTEMP.

From the beginning, I reset the operation to prevent from initial operation result makes error to our new operation. After it, I click on the LOADTEMP present in the right side of the program in available instruction and this will be selected and I click on the address 0 to contains LOADTEMP in address 0 container. Similarly, I do the same for every available instruction that I need to put in the memory address in sequence. Completing the above process, I run the program and at first the program asks me for LOADLOOP and I entered 20 in it which means my program can add 20 numbers before terminate. After inserting the value in LOADLOOP, the memory address further goes to memory address 1 which contains the STRTBLK for LOADLOOP. Moving further in the address 2 container which request number from the user and I as a user I provide 4 at first time. This number is added using the ADD which is located in memory address 3 and DISPACC will store the value after addition. After the DISPACC in address 4, we can see ENDBLK which defines the ending point of the loop. The program between STRTBLK and ENDBLK runs 20 times requesting 20 numbers from the user and store it and repeating the same process again and again. In my case, the entered number in LOADTEMP is (4, 1,5, 3, 2, 10, 6, 1, 5, 12, 8, 7, 3, 25, 19, 14, 5, 8, 7, 4). This numbers are added together and I get the output of 149 in Output box after the program is HALT.

i. Write a program that will let you multiply 2 numbers together. List your program below and explain how it works (35 marks)

LOADTEMP

ADD

LOADLOOP

STRTBLK

ADD

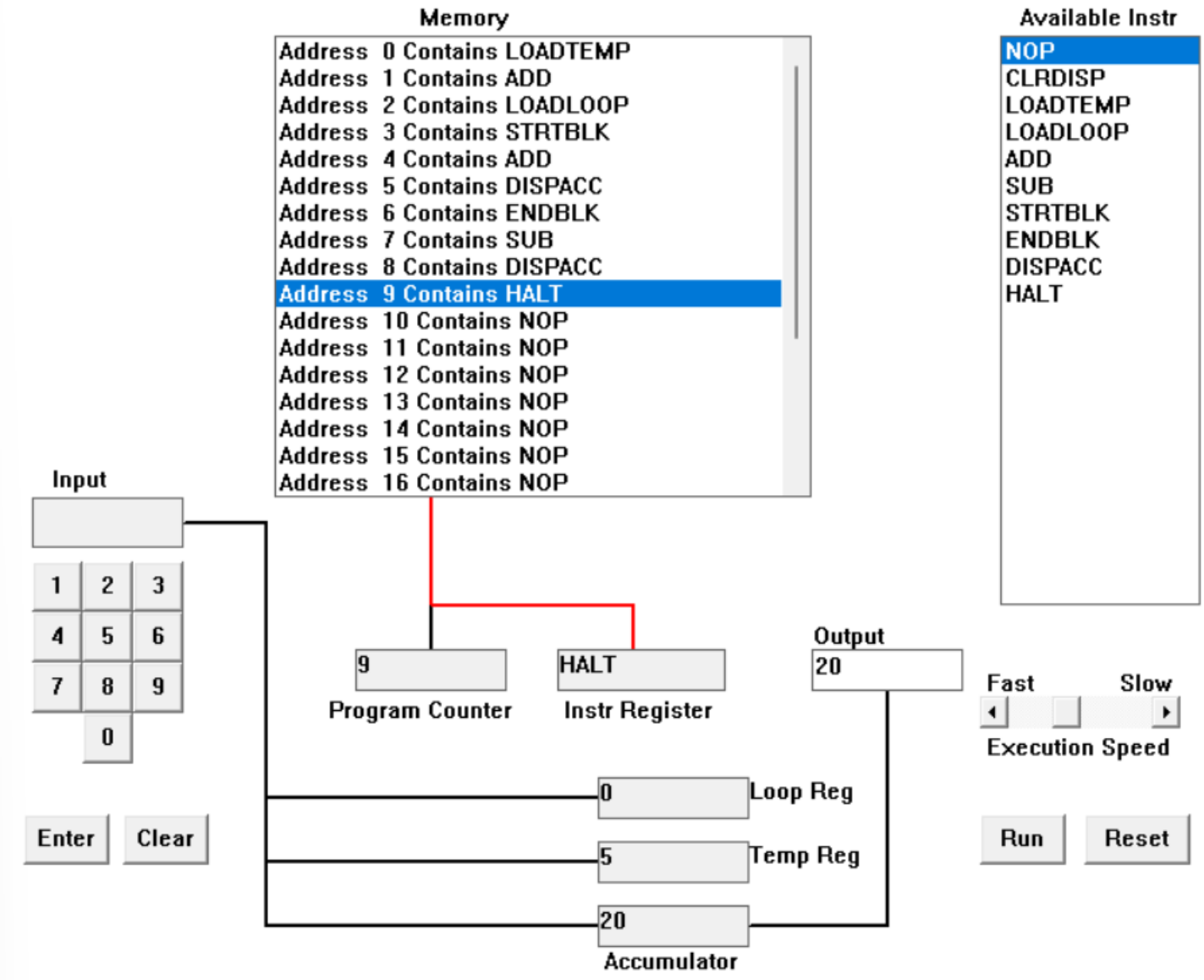
DISPACC

ENDBLK

SUB

DISPACC

HALT




The above program will help us to get multiply of 2 numbers together. In this program I use the instruction like LOADTEMP, ADD, LOADLOOP, STRTBLK, DISPACC, ENDBLK, SUB and HALT. LOADTEMP is used to get a number form the keypad, completed by the Enter key, into the Temporary Register. ADD

is used to add the temporary register to the accumulator. LOADLOOP is used to get a number from the keypad, completed by the Enter key, into the Loop Register. STRTBLK is the starting block of loop, DISPACC is used to display the contents of the Accumulator, ENDBLK is used to determine the ending block of loop, SUB is used to subtract the temporary register from the Accumulator and HALT is used to stop the program.

Let's visualize from the beginning, at first LOADTEMP is goes to the instruction register and it ask the user to get the number and after giving number by the user it will register to temporary register. After it, Instruction Register get another memory address value which is ADD, this number will be added to the number added in temporary register. After addition, the memory address shifted to address no 2 which request looping number, it will be register to loop register. After it, instruction register takes next instruction of memory address which has the instruction of STRTBLK which determines the starting of block. Inside the loop, we have ADD and DISPACC which add the temporary register value and store it into accumulator. After it we have ENDBLK in the address 6 container of memory which defines the end of loop. After that we have the SUB which is used to subtract the temporary register value to accumulator as our loop gives extra one loop. So, I use subtract to get the correct value and DISPACC is used to store the value in accumulator and at last HALT stop the program and gives us the output in the output box. We can see that this program run 9 times in program counter without including the loop.

In my case, for LOADTEMP, I provided the number 5 from the input field which register into temporary register. It further adds and I provide the value of number 4 which register into loop register. Inside the loop block, we get +5 each time in DISPACC for 4 times which return the value 25 but we know that 5*4 will be 20. So, I use SUB to subtract the Temporary register value from accumulator values which gives 20 in Accumulator in Address 8 DISPACC. At last, I terminate the program using HALT and display the output 20 in output box.