4CS015 Fundamentals of Computing – Workshop-6

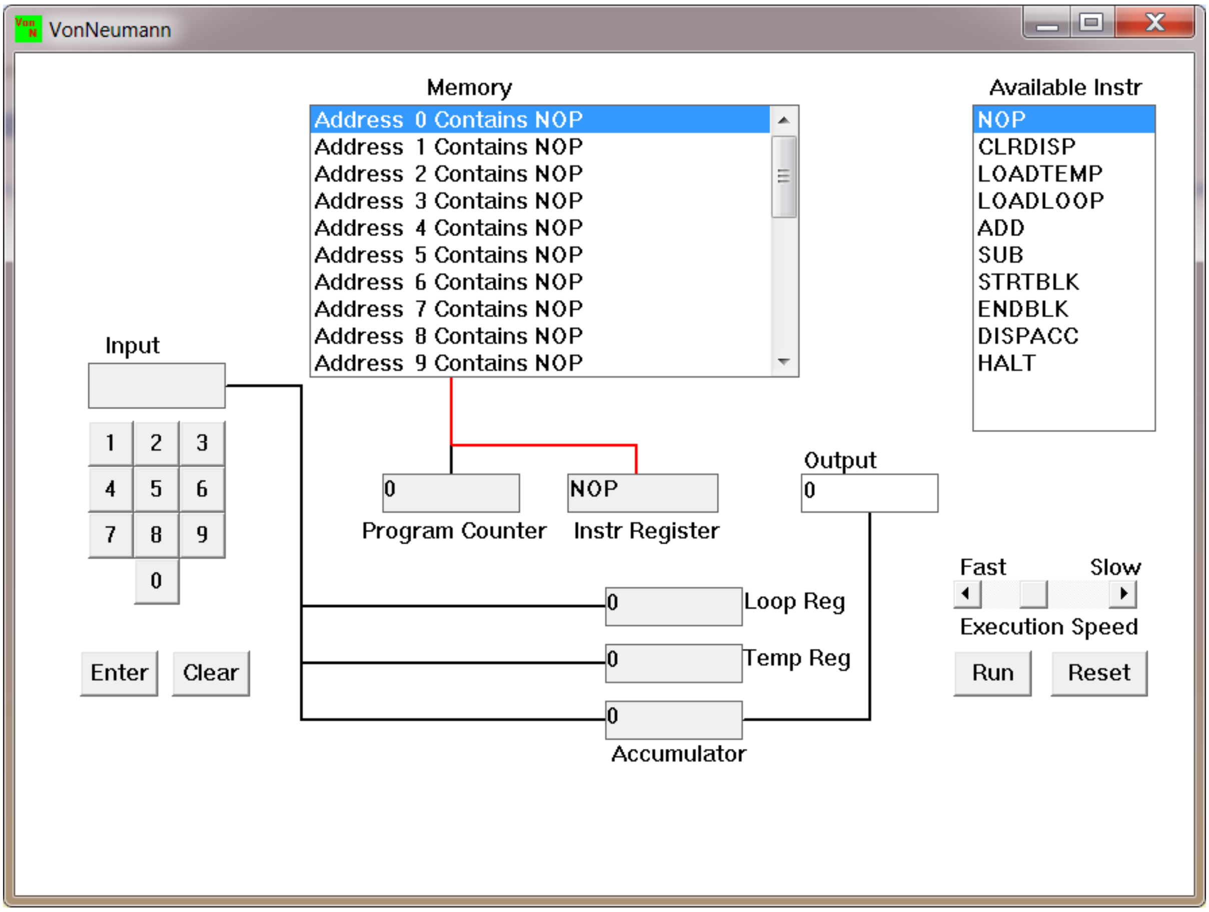
NAME: Dikshya Gurung

Student ID: 2418265

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:



```
Address  0 Contains NOP
Address  1 Contains NOP
Address  2 Contains NOP
Address  3 Contains NOP
Address  4 Contains NOP
Address  5 Contains NOP
Address  6 Contains NOP
Address  7 Contains NOP
Address  8 Contains NOP
Address  9 Contains NOP
```

Program Counter: 0

Instr Register: NOP

Output: 0

Loop Reg: 0

Temp Reg: 0

Accumulator: 0

The simulator has a small program memory area which is available for programming. To enter your program instructions 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 instructions 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.

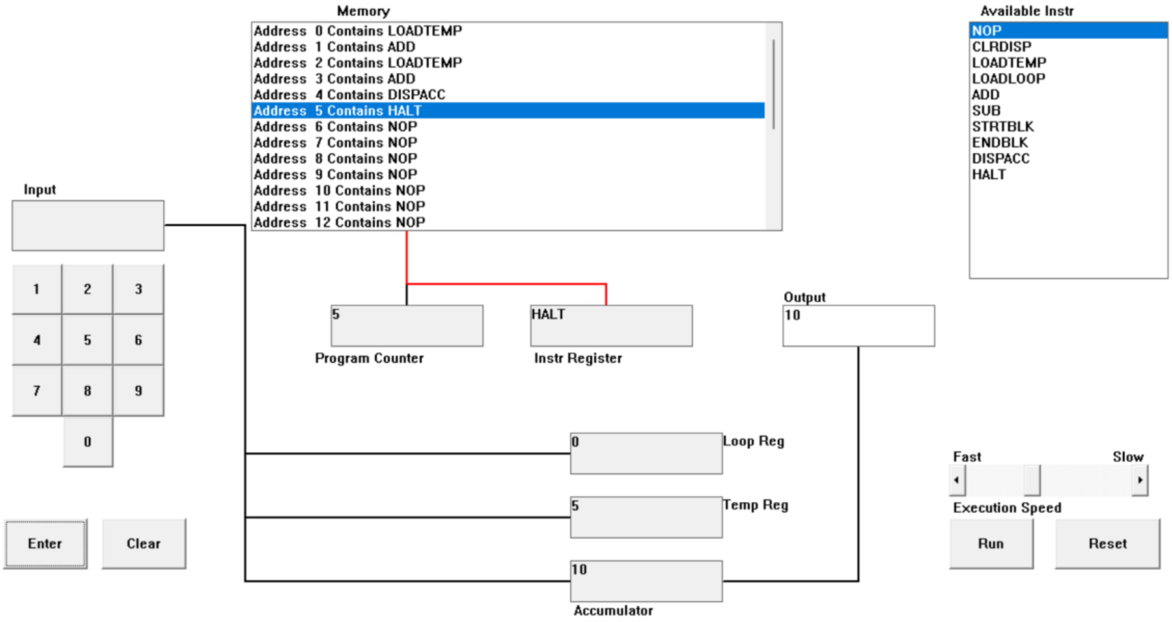
Click 2 or 3 numbers on the keypad and then click the “Enter” button. The simulator will then resume running the program and execute the instruction “ADD”. This adds the number that you just entered, to the zero in the accumulator.

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 answer below (10 marks)



Memory

```
Address  0 Contains LOADTEMP
Address  1 Contains ADD
Address  2 Contains LOADTEMP
Address  3 Contains ADD
Address  4 Contains DISPACC
Address  5 Contains HALT
Address  6 Contains NOP
Address  7 Contains NOP
Address  8 Contains NOP
Address  9 Contains NOP
Address  10 Contains NOP
Address  11 Contains NOP
Address  12 Contains NOP
```

Program Counter: 5

Instr Register: HALT

Output: 10

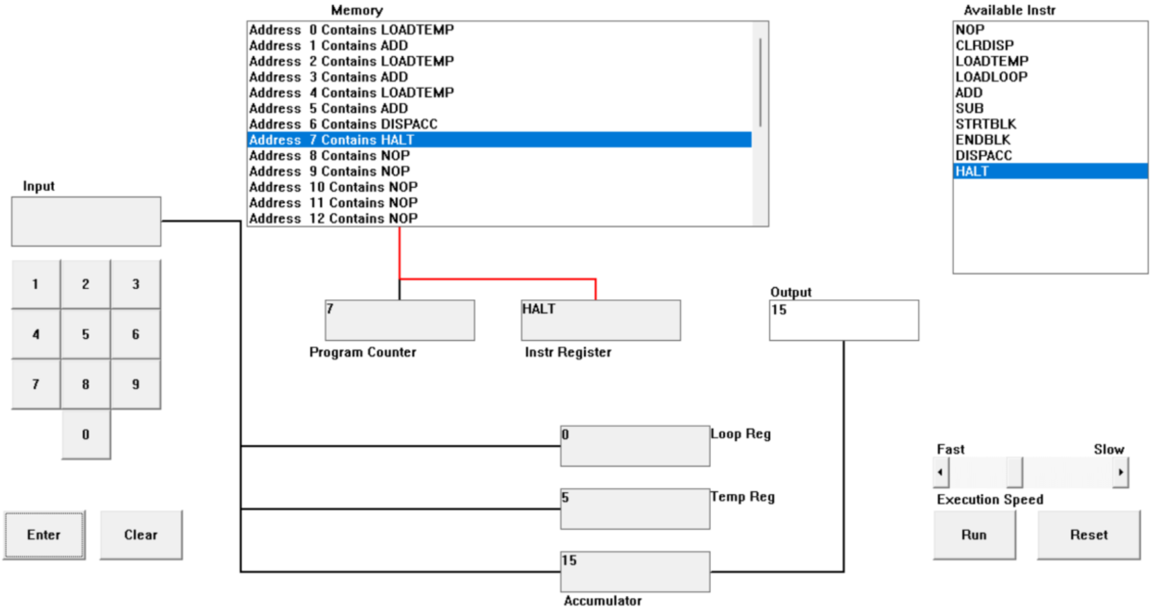
Loop Reg: 0

Temp Reg: 5

Accumulator: 10

-> This program appears to perform two values from a temporary location, add them together display the result and halt. At first, LOADTEMP is stored at memory address 0 and then in memory address 2. Similarly, ADD is stored at memory address 1 and memory address 3 as shown on the above diagram. Then at memory address 4 and 5 DISPACC and HALT is used sequentially. The LOADTEMP loads a value from a temporary location into memory, ADD performs an addition operation, DISPACC displays the result of the addition operation where as HALT halts the program execution.

e. Write a program to add 3 numbers together. List your program below (10 marks)



Memory

```
Address  0 Contains LOADTEMP
Address  1 Contains ADD
Address  2 Contains LOADTEMP
Address  3 Contains ADD
Address  4 Contains LOADTEMP
Address  5 Contains ADD
Address  6 Contains DISPACC
Address  7 Contains HALT
Address  8 Contains NOP
Address  9 Contains NOP
Address  10 Contains NOP
Address  11 Contains NOP
Address  12 Contains NOP
```

Program Counter: 7

Instr Register: HALT

Output: 15

Loop Reg: 0

Temp Reg: 5

Accumulator: 15

LOADTEMP

ADD

LOADTEMP

ADD

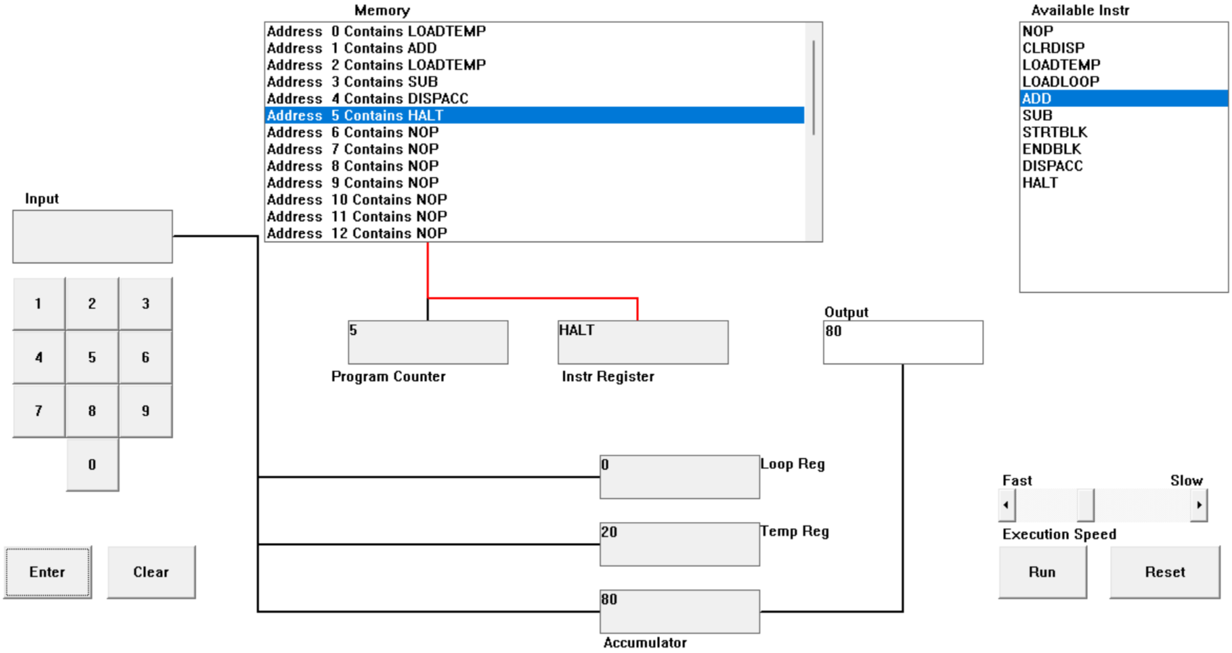
LOADTEMP

ADD

DISPACC

HALT

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



Memory

```
Address  0 Contains LOADTEMP
Address  1 Contains ADD
Address  2 Contains LOADTEMP
Address  3 Contains SUB
Address  4 Contains DISPACC
Address  5 Contains HALT
Address  6 Contains NOP
Address  7 Contains NOP
Address  8 Contains NOP
Address  9 Contains NOP
Address  10 Contains NOP
Address  11 Contains NOP
Address  12 Contains NOP
```

Available Instr

```
NOP
CLRDISP
LOADTEMP
LOADLOOP
ADD
SUB
STRTBLK
ENDBLK
DISPACC
HALT
```

Program Counter: 5

Instr Register: HALT

Output: 80

Loop Reg: 0

Temp Reg: 20

Accumulator: 80

LOADTEMP

ADD

LOADTEMP

SUB

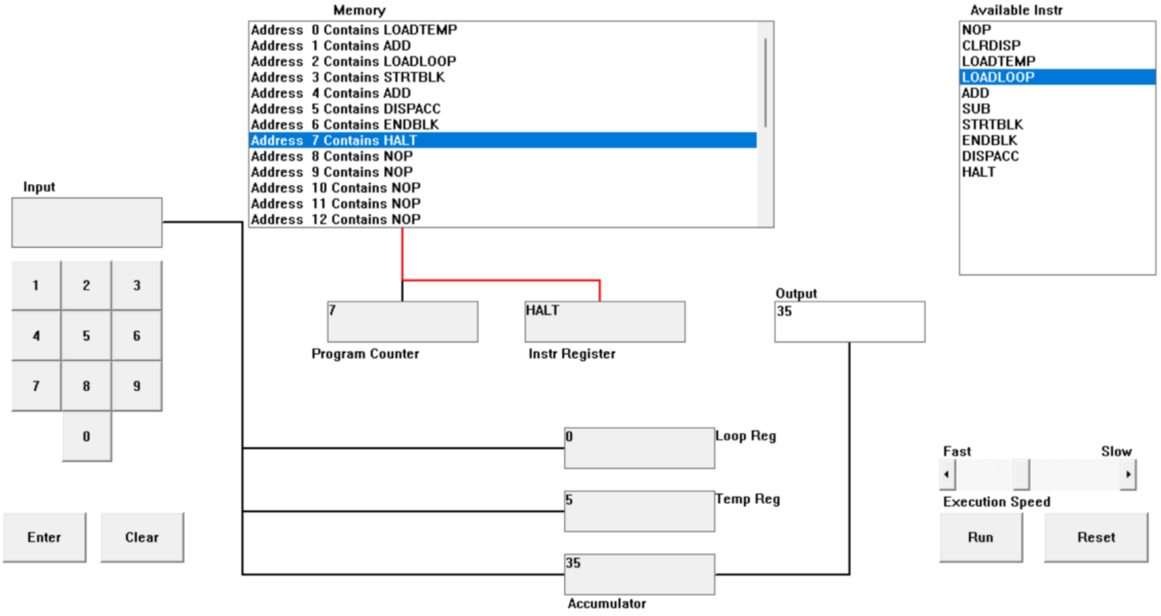
DISPACC

HALT

g. Load the following program into the simulator:

LOADTEMP
ADD
LOADLOOP
STRTBLK
ADD
DISPACC
ENDBLK
HALT

Run it and when it reach 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 your answer below in the form of an equation (10 marks)



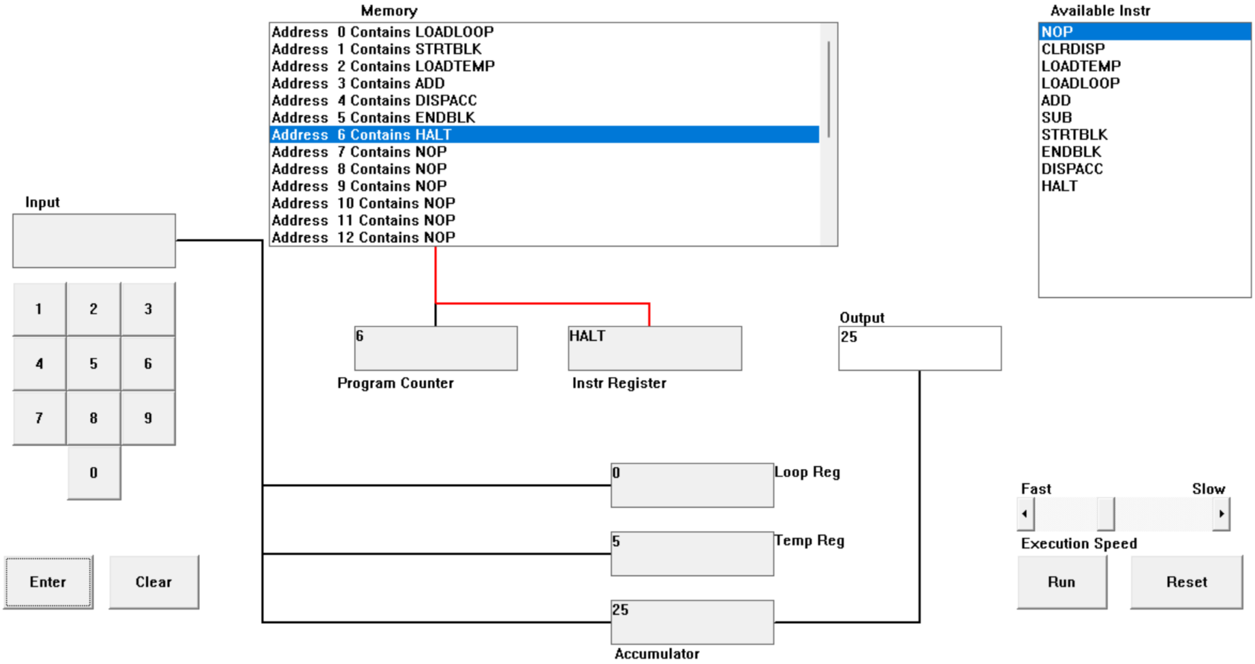
```
Address  0 Contains LOADTEMP
Address  1 Contains ADD
Address  2 Contains LOADLOOP
Address  3 Contains STRTBLK
Address  4 Contains ADD
Address  5 Contains DISPACC
Address  6 Contains ENDBLK
Address  7 Contains HALT
Address  8 Contains NOP
Address  9 Contains NOP
Address  10 Contains NOP
Address  11 Contains NOP
Address  12 Contains NOP
```

Program Counter: 7
Instr Register: HALT
Output: 35
Loop Reg: 0
Temp Reg: 5
Accumulator: 35

-> Here this program is used to run a loop that runs 6 times as we enter 6 on LOADLOOP and adds 5 five times inside the loop. STRTBLK is used to start loop block and ENDBLK is used to end loop block. while the loop runs value from the temp register is added to the value from the accumulator and Their sum is stored at the accumulator again which is again added as the loop runs.

Equation : 5+6*(5) = 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)



Memory

```
Address  0 Contains LOADLOOP
Address  1 Contains STRTBLK
Address  2 Contains LOADTEMP
Address  3 Contains ADD
Address  4 Contains DISPACC
Address  5 Contains ENDBLK
Address  6 Contains HALT
Address  7 Contains NOP
Address  8 Contains NOP
Address  9 Contains NOP
Address  10 Contains NOP
Address  11 Contains NOP
Address  12 Contains NOP
```

Available Instr

```
NOP
CLRDISP
LOADTEMP
LOADLOOP
ADD
SUB
STRTBLK
ENDBLK
DISPACC
HALT
```

Program Counter: 6

Instr Register: HALT

Output: 25

Loop Reg: 0

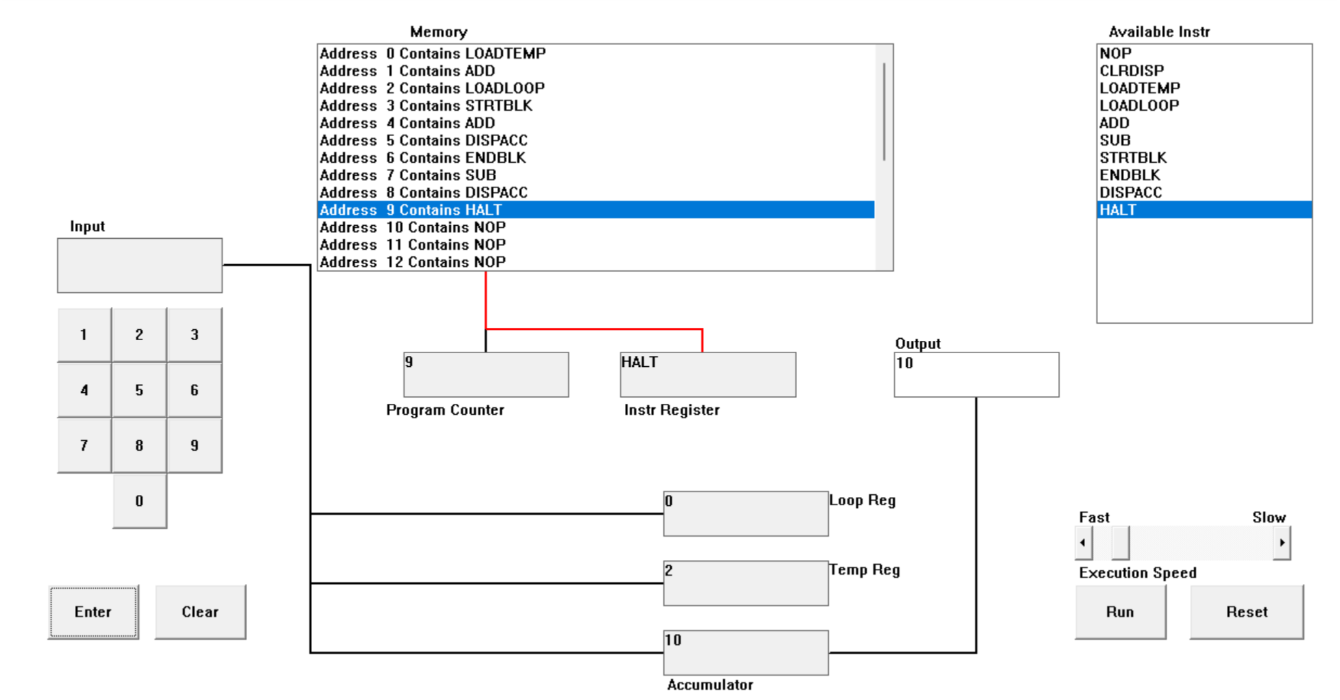
Temp Reg: 5

Accumulator: 25

-> In this program, we can add multiple numbers together. Here LOADLOOP accepts input as how many times the loop will run. Then STRTBLK starts loop block where user input is taken and addition operator is added to it. DISPACC is used to display output. This loop block will run as many times as user has entered the value of LOADLOOP. Then ENDBLK ends the loop and HALT ends entire program.

In the above diagram at LOADLOOP input 5 is given, which indicates the user can input 5 numbers and the program return their sum.

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



Memory

```
Address  0 Contains LOADTEMP
Address  1 Contains ADD
Address  2 Contains LOADLOOP
Address  3 Contains STRTBLK
Address  4 Contains ADD
Address  5 Contains DISPACC
Address  6 Contains ENDBLK
Address  7 Contains SUB
Address  8 Contains DISPACC
Address  9 Contains HALT
Address  10 Contains NOP
Address  11 Contains NOP
Address  12 Contains NOP
```

Program Counter: 9

Instr Register: HALT

Output: 10

Loop Reg: 0

Temp Reg: 2

Accumulator: 10

-> In this program LOADTEMP is used to take the value of multiplicand and LOADLOOP is used to take the value of the multiplicator. The program contains various instructions; those instructions performs in the following process:

LOADTEMP - it takes input from the user which is multiplicand and stores it at a temporary register

ADD - it adds an addition operator to load the temp value

LOADLOOP - takes a value that denotes how many times the loop will run. Its value also can be called

Multiplicator - Its value is stored at the loop Register and it decreases as the loop runs.

STRTBLK- It denotes the start of the loop block.

ADD - It adds the value of the temporary register with the accumulator

DISPACC - It displays the Output

ENDBLK – It denotes End of loop block

SUB – It subtracts multiplicand from output

DISPACC – It displays output

HALT – It ends the program