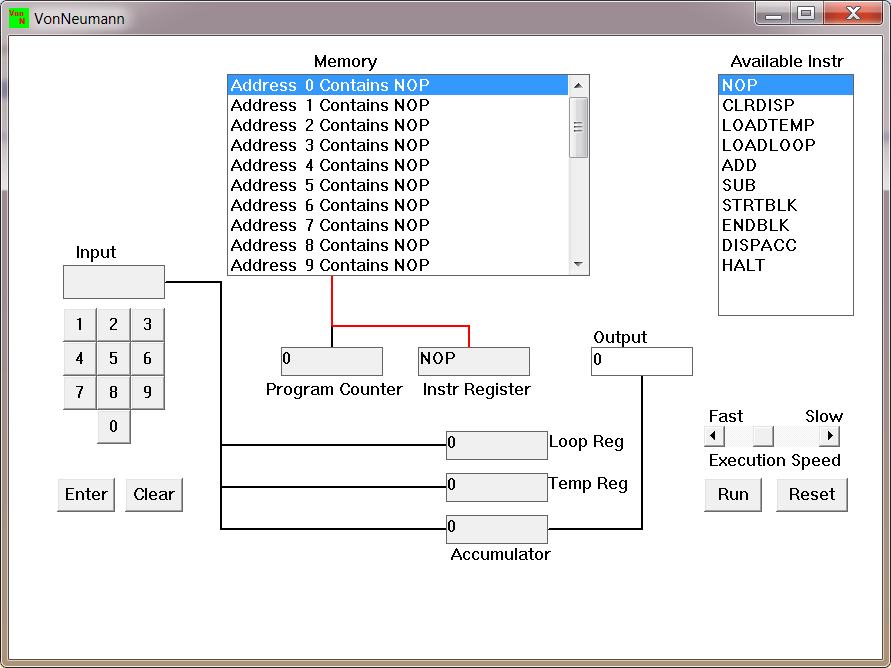
# 4CS015 Fundamentals of Computing – Workshop-6

**Workshop tasks:**

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



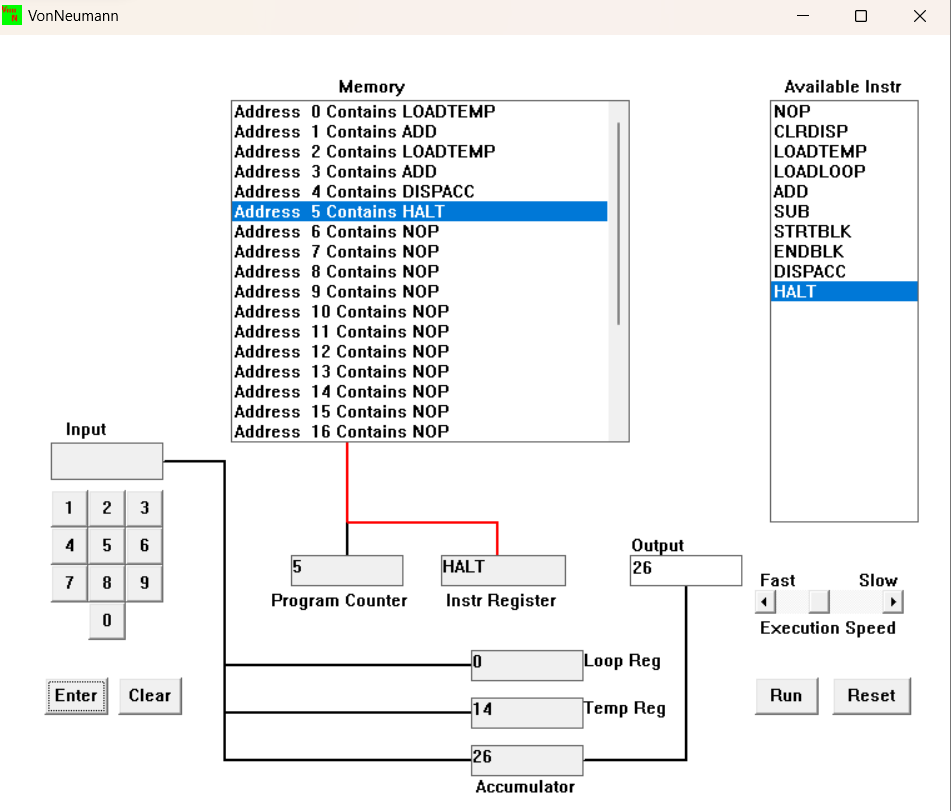
This simulator understands only the following ten instructions:

|  |  |
| --- | --- |
| 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 |

* 1. Load the following program into the memory:  
     LOADTEMP  
     ADD  
     DISPACC  
     HALT  
       
     To begin, select "LOADTEMP" from the list of directions to the right of the simulator window. Click on the memory region that says "Number 0 Holds NOP" afterwards when. Then it becomes "Address 0 Contains LOADTEMP." Proceed in this manner with "Address 1" and so on until the entire program has loaded.
  2. 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.   
       
     Before pressing the "Enter" button, enter two or three digits on the keypad. The "ADD" command will then be carried out by the simulator once the program resumes executing. To the zero in the accumulator, this adds the number you just supplied.

"DISPACC," which stands for "Display Accumulator," is the following command, and it does this. When the command "HALT" is executed after that, the simulator terminates the program.

* 1. 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)

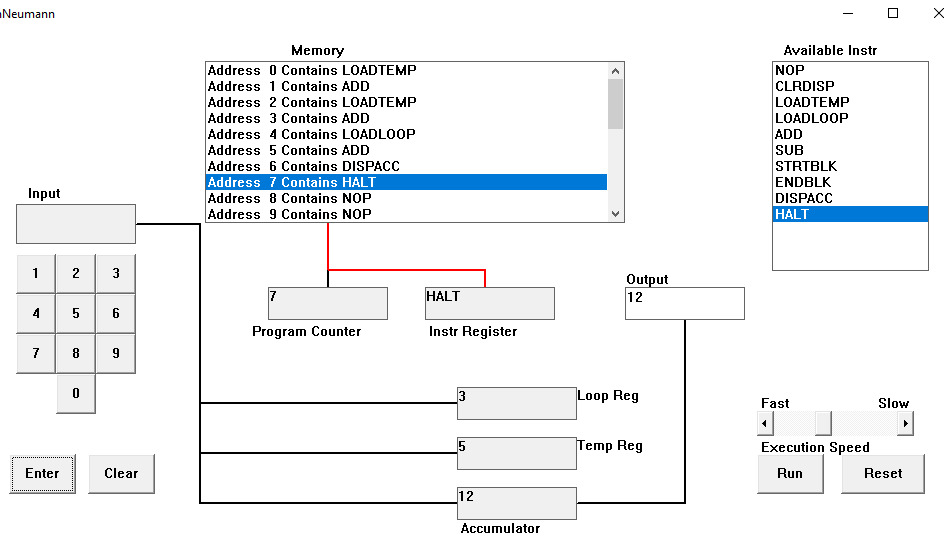


Simple mathematical calculations are carried performed by the program using assembly code. It starts by reading a value from a temporary memory location and loading it into the accumulator register. The result is then saved in a temporary memory address after the ADD instruction has been used to add a new value to the one already stored in the accumulator. This step is repeated in order to carry out the addition of the two values. The software utilizes the DISPACC instruction to show the addition's result on the screen and the HALT command to halt operation.

It is important to keep in mind that this software most likely functions as a part of a bigger system because it cannot control the input and output of values. The source of the values used in the computations is unknown, however they may be set in the software or given by an external source. In addition, the application is not designed to manage potential operational issues like overflow issues.

To make the program more useful and trustworthy, other features would need to be included, such as input and output capabilities to take user input and error handling tools to address any potential problems during operation.

Write a program to add 3 numbers together. List your program below (10 marks)  
  
  
  
With the help of the instruction LOADTEMP NUM1, the value from memory address NUM1 is loaded into the temporary register.



When using the ADD instruction, the value previously stored in the accumulator is added to the value stored in the temporary register.

The LOADTEMP NUM2 instruction loads the value present at memory address NUM2 into the temporary register.

When using the ADD instruction, the value previously stored in the accumulator is added to the value stored in the temporary register.

The value at memory location NUM3 is loaded into the temporary register by this command.

When using the ADD instruction, the value previously stored in the accumulator is added to the value stored in the temporary register.

DISPACC: The output of the addition is displayed on screen using this command.

The program's execution is stopped using the HALT instruction.

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

ADD

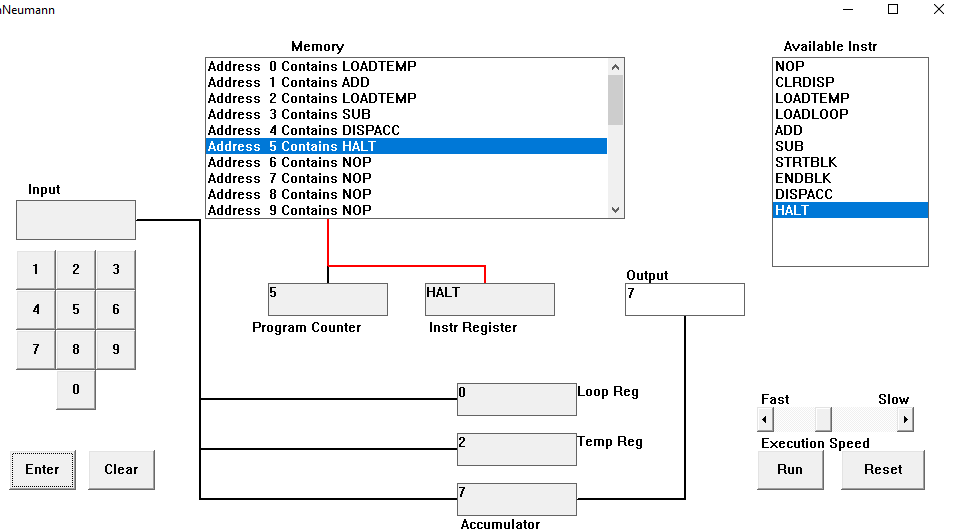
LOADTEMP

SUB

DISPACC

HALT

While LOADTEMP demands a value, the input is added to the temp register's number and sent to the accumulator. When LOADTEMP requests an input number once more, the input value will once more be sent to the temp register, where the function SUB will subtract the second number from the first number that was supplied in the first and then pass the result to the accumulator before the solution is displayed and the program is terminated. In conclusion, this program provides a rapid and effective way to subtract two integers.



* 1. 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)  
       
     LOADTEMP

ADD

LOADLOOP

STRBLK

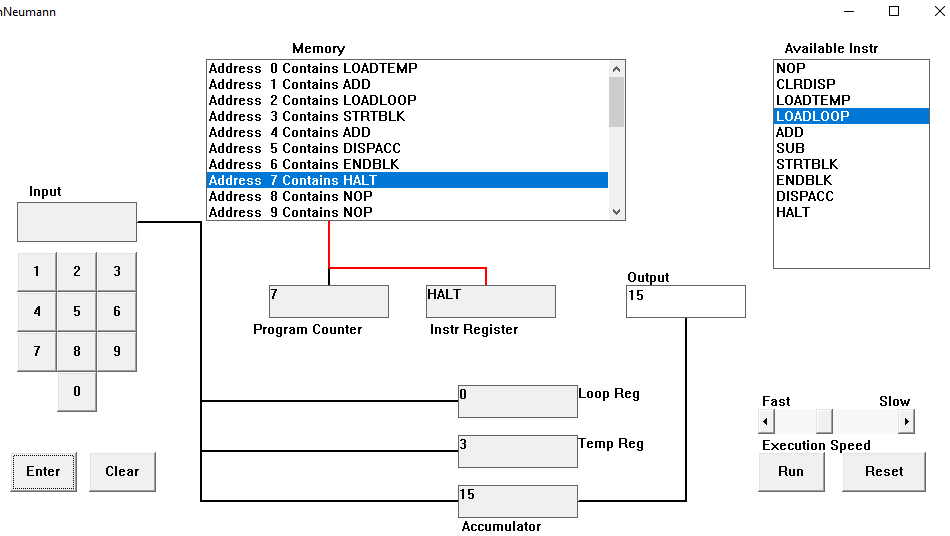
ADD

DISPACC

ENDBLK

HALT

After pressing the run button, the LOADTEMP prompts for input, and the program is then started. The input is then added with the number of temp registers and delivered to the accumulator, after which a loop runs for as long the user desired. The STRTBLK command initiates the loop, which is then terminated by the ENDBLK instruction after one iteration in which the value that was added to the user's previous value input is shown.



* 1. 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)  
       
     LOADLOOP

STRBLK

LOADTEMP

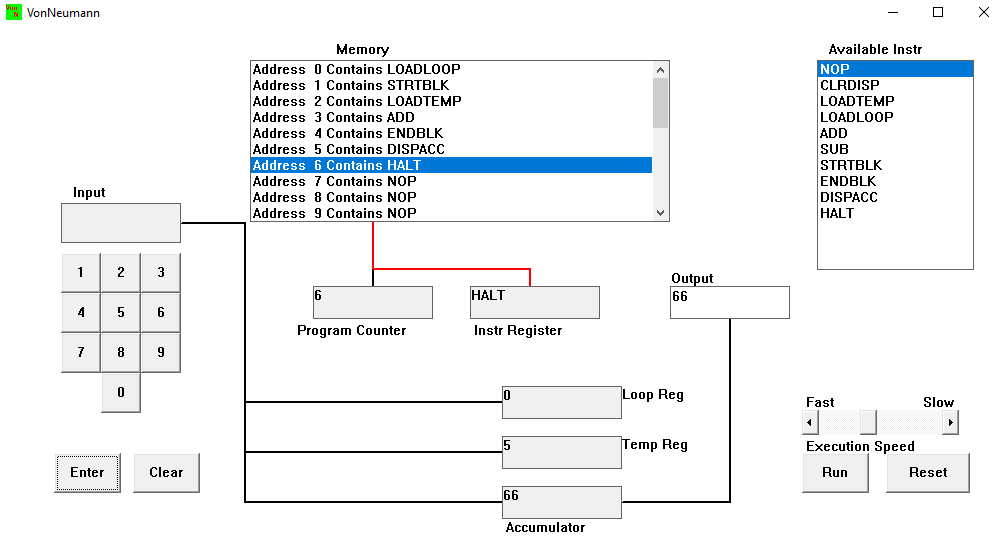
ADD

ENDBLK

DISPACC

HALT

This program allows the user to add 5, 10, or 20 numbers together by utilizing the commands LOADLOOP, STRBLK, LOADTEMP, ADD, ENDBLK, DISPACC, and HALT. LOADLOOP initiates a loop that loads the integers into memory one at a time while STRBLK is used to determine the memory's starting address. The amount is then added to the total using ADD after being loaded from memory to the accumulator using LOADTEMP. After using ENDBLK to end the loop and DISPACC to display the sum, the program ends.



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

LOADLOOP

STRTBLK

ADD

ENDBLK

DISPACC

HALT

With the use of this program, users may multiply two numbers together. Using the LOADTEMP and LOADLOOP commands, the program first stores the first value before storing the second one. The STRTBLK instruction is then used to indicate the start of the program block and the ADD command is used to add the two numbers. The ENDBLK command is then used to indicate that the program block has come to an end, and the DISPACC command is then used to show the results. After that, the program is terminated using the HALT command. In general, this software makes it simple for users to multiply two numbers together, making it a helpful tool for doing basic mathematical operations.

