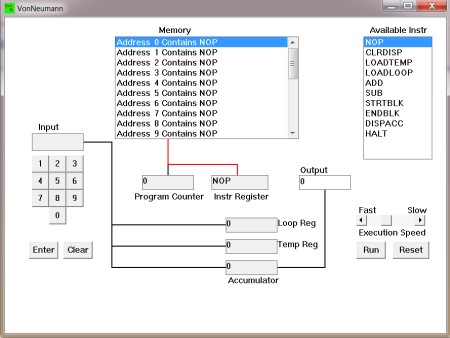
**4CS015 Fundamentals of Computing – Workshop-6**

**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 theWeek 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.

|  |
| --- |
| Get a number from the keypad, completed  LOADTEMP  by the Enter key, into the Temporary Register. |
| LOADLOOPGet a number from the keypad, completed by the Enter key, into the Loop Register. |
| CLRDISP Clear the Display. |
| ADDAdd the Temporary Register to the Accumulator |
| SUBSubtract 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 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.

1. Run the program by clicking on the “Run” button. The simulator would highlight theAddress 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”.

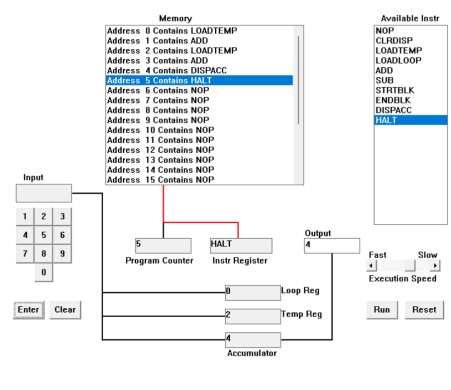
1. Load the following program into the simulator:

LOADTEMP ADD

LOADTEMP

ADD

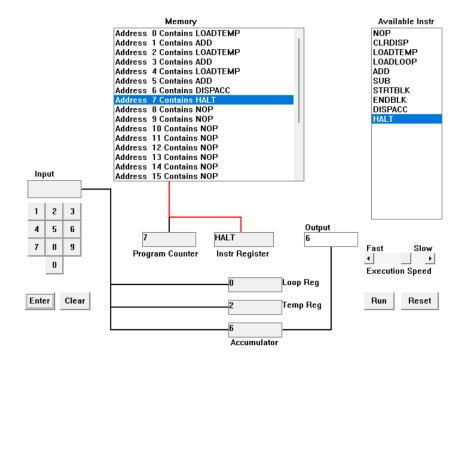
DISPACC HALT



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

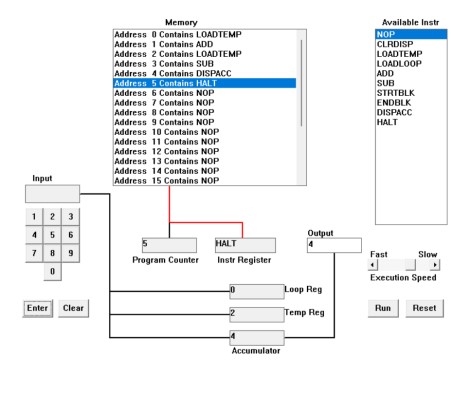
This program can add 2 numbers. The "loadtemp" function takes input from the user, and the "add" function performs the addition. After "loadtemp" and "add," it displays the final result. The "halt" function terminates the program.

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



This program can add 3 numbers. The "loadtemp" function takes input from the user, and the "add" function performs the addition. After "loadtemp" and "add," it displays the final result. The "halt" function terminates the program.

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



This program can subtract 2 numbers. The "loadtemp" function takes input from the user, and the "sub" function performs the subtraction of the two numbers given by the user. After that, the "dispacc" function displays the final result. The "halt" function terminates the program.

1. Load the following program into the simulator:

LOADTEMP

ADD

LOADLOOP

STRTBLK ADD

DISPACC

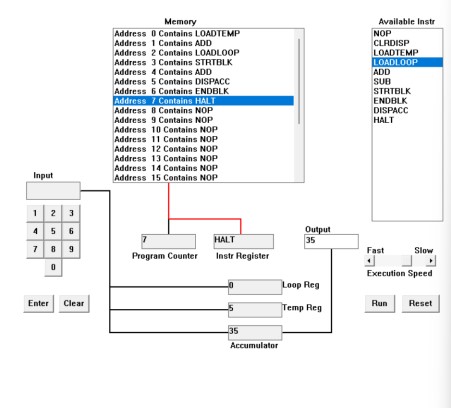
ENDBLK

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)



This program allows the user to input a number and then repeatedly add that number using a loop. The user inputs a number using the "loadtemp" function, and the loop is initiated by the "loadloop" function. Within the loop, the number input by the user is added to itself the specified number of times. The program concludes when the loop has completed, for example, in this case 5+(5\*6)

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

LOADLOOP

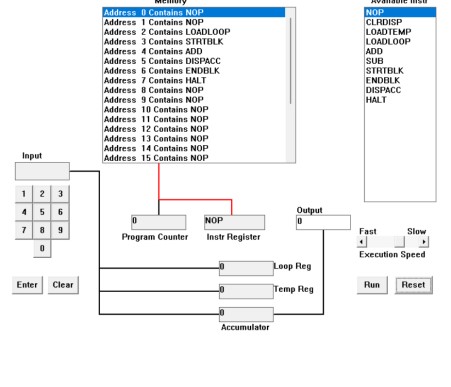
STRTLOOP

ADD

DISPACC

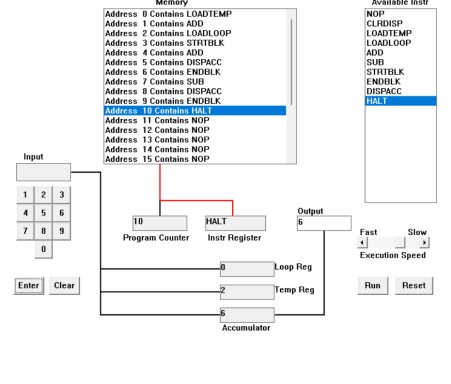
ENDBLK

HALT



This program allows the user to input a number and then repeatedly add that number using a loop. The user inputs a number using the "loadtemp" function, and the loop is initiated by the "loadloop" function. Within the loop, the number input by the user is added to itself. The program concludes when the loop has completed.

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



LOADTEMP

ADD

LOADLOOP

STRTBLK

ADD DISPACC

ENDBLK

SUB

DISPACC

ENDBLK

HALT

This program enables the user to input two numbers and then multiply them together.

The user's input is taken and multiplied using the "add" and "loadtemp" functions. The user then inputs the number of times the first and second number should be multiplied in the "loadloop" stage. For example, if the inputs for "loadtemp" and "loadloop" are 2 and 3 respectively, the program will multiply 2 by 3. The program then proceeds to the "sub" function after the looping is completed which subtracts one from the output. The "sub" function is vital for ensuring the program generates the accurate output, as it prevents the program from repeating the looping of the input number one more time.