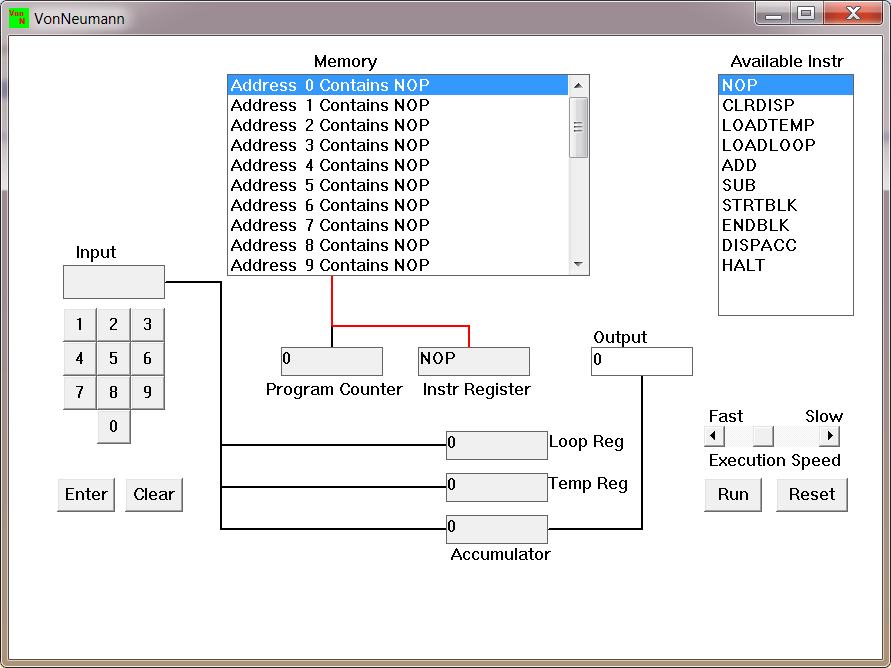
# Name: Akriti Kumari Dev

# 4CS015 Fundamentals of Computing – Workshop-6

**Workshop tasks:**

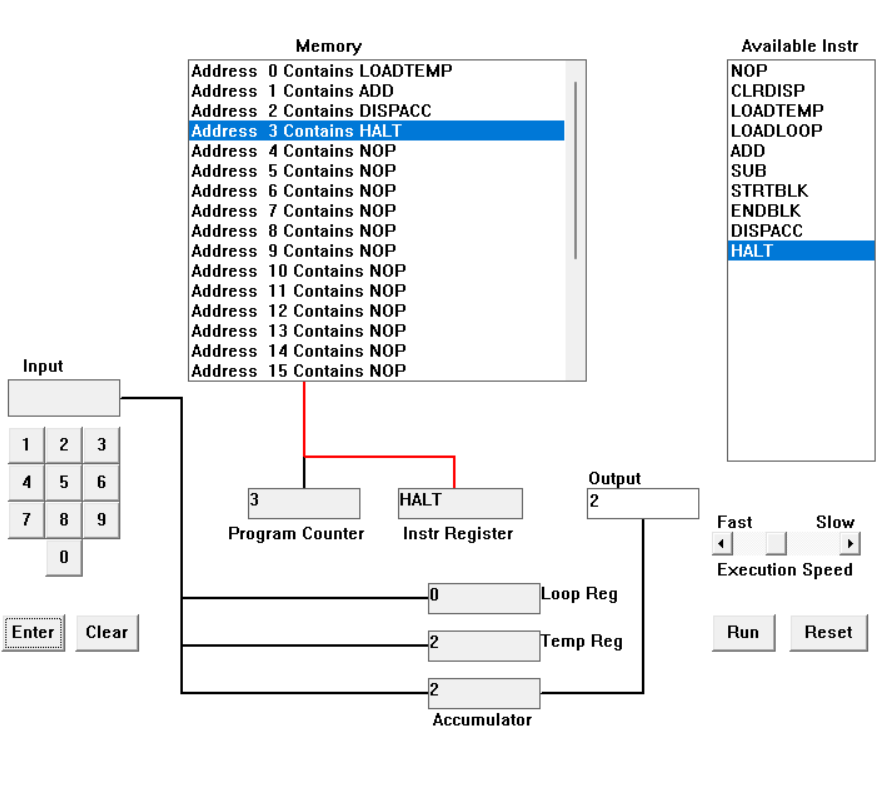
1. Von Neumann Simulator. This program simulates a very simple computer with the von Neumann architecture.
   1. Download the von Neumann Simulator (VonNeumann.exe) program from WOLF in the Week 5 folder. Save it in your Documents folder and run it. You will see a window similar to this:

  
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 |

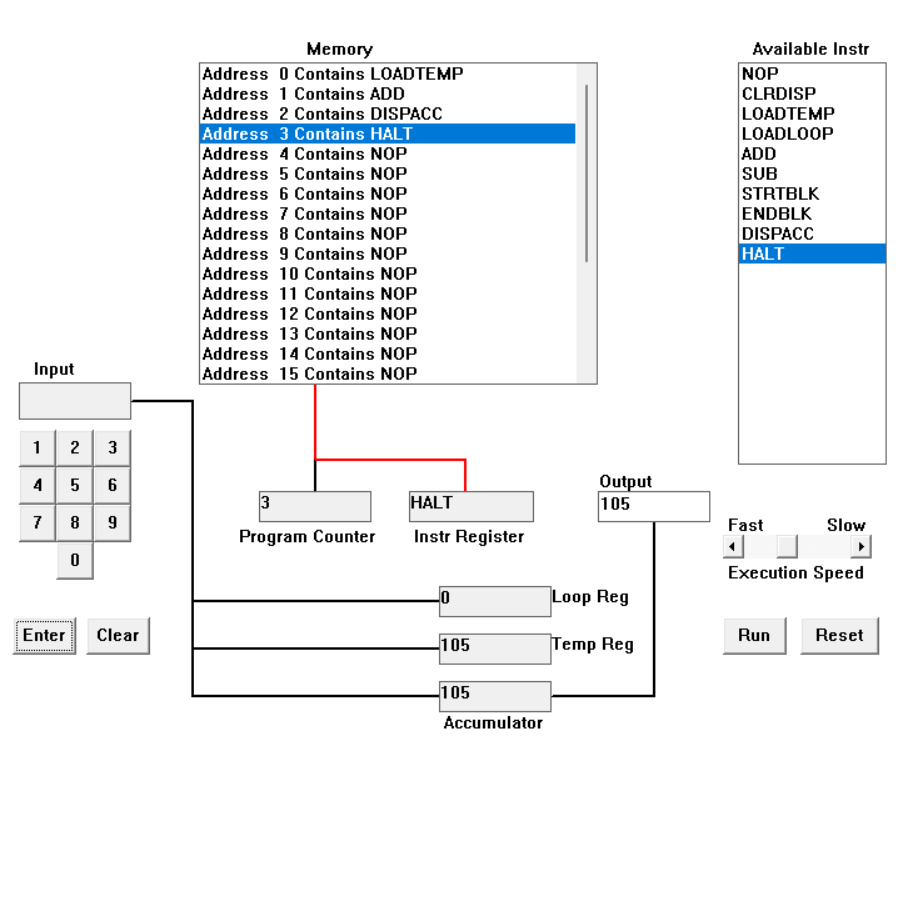
* 1. Load the following program into the memory:  
     LOADTEMP  
     ADD  
     DISPACC  
     HALT  
       
     To do this, first click on the “LOADTEMP” in the list of instructions on the right of simulator window. Then click on Memory location with “Address 0 Contains NOP”. This will then change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



LOADTEMP (2)  
 ADD (2)  
 DISPACC (2)  
 HALT

Using the LOADTEMP (2) instruction, a temporary register is loaded with the value 2. The value currently kept in the temporary register is increased by the value 2 when using the ADD (2) instruction. The temporary register is holding the value 0+2=2, which is displayed by the DISPACC (2) instruction.  Programming activity is halted by the HALT instruction. In its entirety, the program loads 2 into a register, adds 2 to 0, displays the result 2, and then terminates.

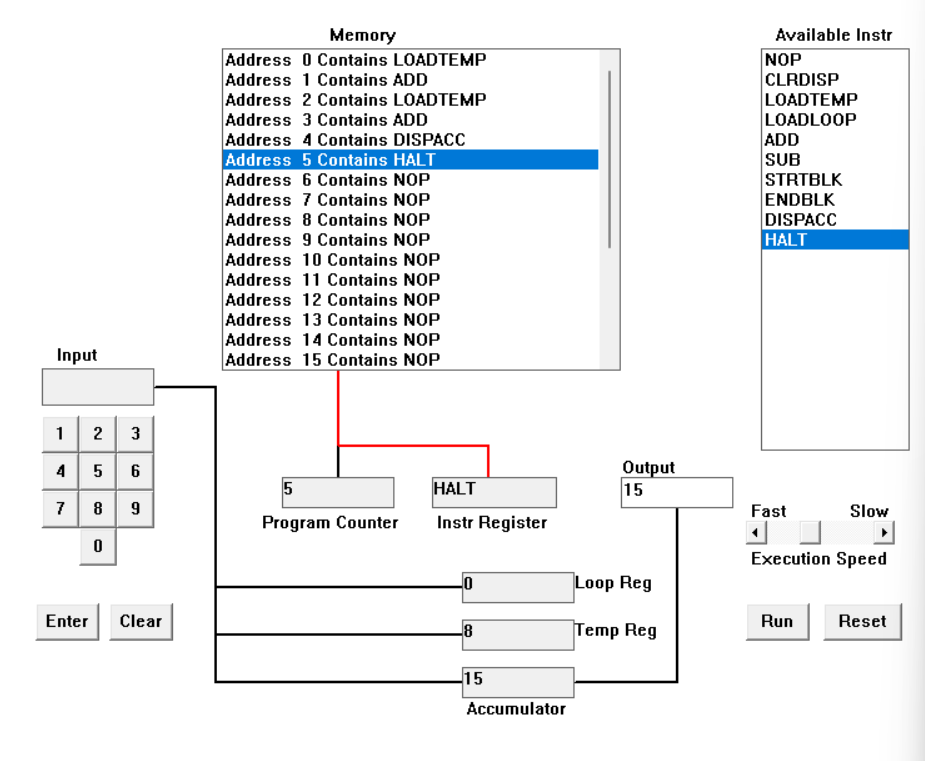
* 1. Run the program by clicking on the “Run” button. The simulator would highlight the Address 0 location and then pause. It is executing the instruction “LOADTEMP” which requires you to input a number into the keypad.   
       
     Click 2 or 3 numbers on the keypad and then click the “Enter” button. 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”.



LOADTEMP (105)  
 ADD (105)  
 DISPACC (105)  
 HALT

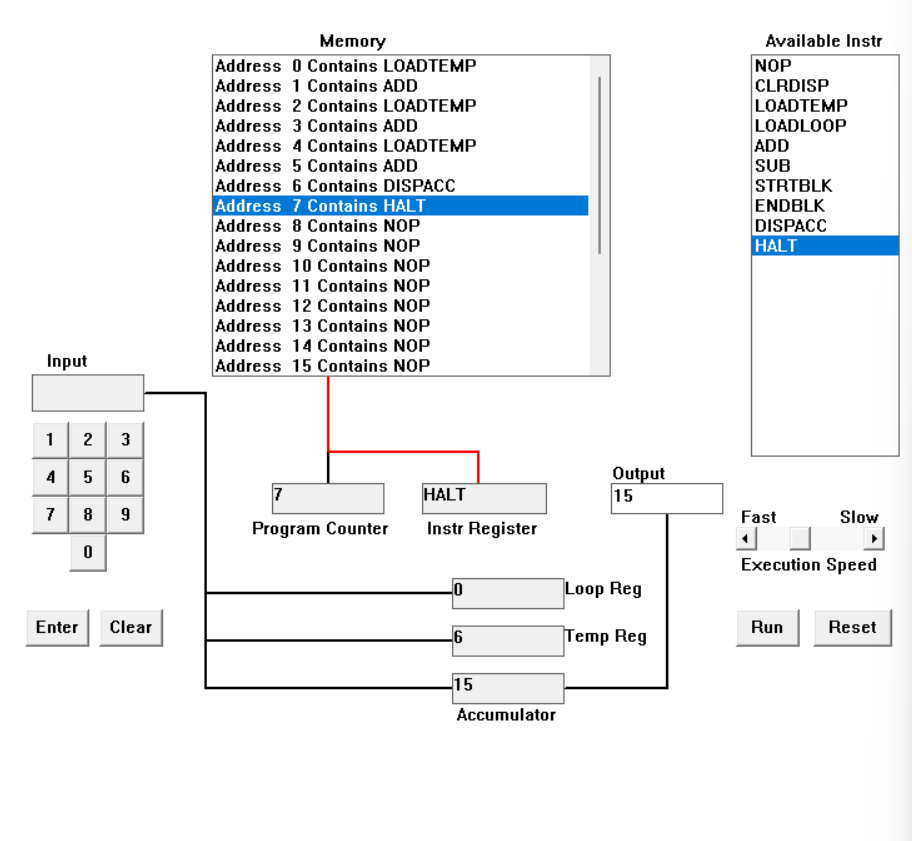
A temporary register is loaded with the value 105 via the LOADTEMP (105) instruction. The temporary register value currently held by the ADD (105) instruction is increased by the value 105. The temporary register's value, which is equal to 0 + 105, is shown by the DISPACC (105) command. The program execution is halted by the HALT instruction. It's important to remember that the value given in the DISPACC instruction (105), which is just used to identify the value being shown, has no bearing on the value that is kept in the register. The value shown is the one that was entered into the register, not the one that was supplied as a parameter. The program loads 105 into a register, adds 105 to 0, displays the result as 105, and then halts.

* 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)

  
LOADTEMP (7)  
ADD (8)  
LOADTEMP (8)  
ADD (15)  
DISPACC (15)  
HALT

The digits 7 and 8 are kept by the software in a temporary register, often known as the temp register. The value in the accumulator register, often known as the ACC, is then increased by the value in the temporary register. The final value of 15 after addition is shown on the screen using the instruction DISPACC (15). The command HALT signals the conclusion of the program. Overall, the program shows the fixed value 15 on the screen, loads the numbers 7 and 8 into a temporary register, adds those numbers to the accumulator, and then terminates.

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

LOADTEMP (4)

ADD (4)

LOADTEMP (5)

ADD (9)

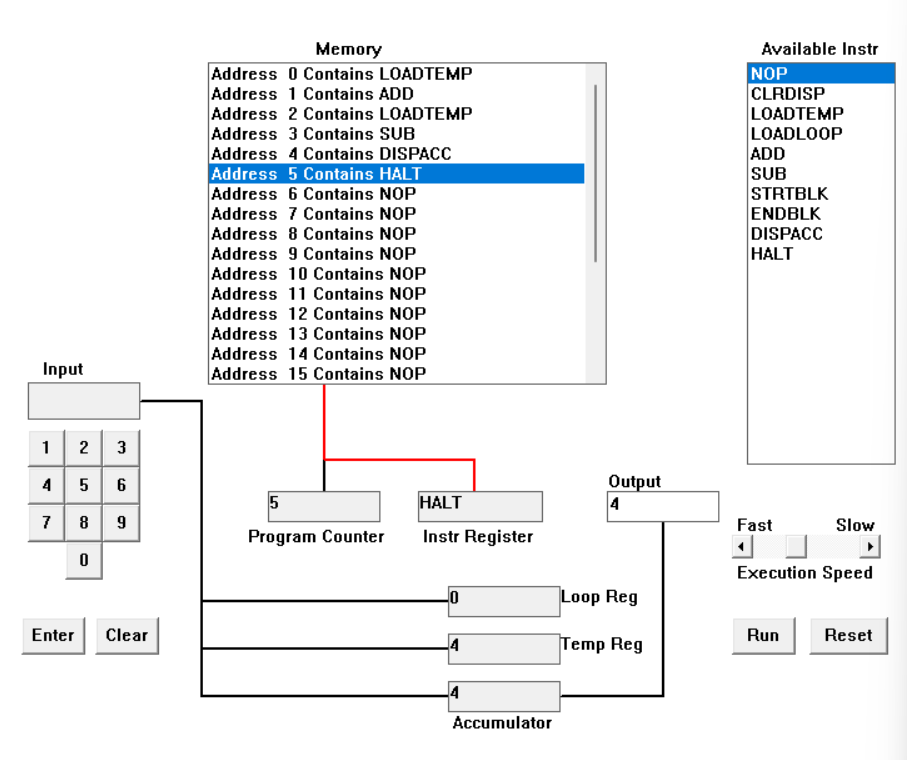
LOADTEMP (6)

ADD (15)

DISPACC (15)

HALT

Using the LOADTEMP instruction, the program first loads the number 4 into a temporary register. The ADD instruction is used to add this value to 4 and put the result in the accumulator. Then, using the LOADTEMP instruction, the program loads the value 5 into the temporary register and adds it to the accumulator's currently holding value of 4. This operation's output, 9, is then saved in the accumulator. Following that, the program uses the LOADTEMP instruction to load the value 6 into the temporary register and add it to the accumulator's current value of 9, which is now 6. The outcome of procedure 15 is next kept in the accumulator. In order to display the value of the accumulator, which is 15, on the screen, the software utilizes the DISPACC instruction. Using the HALT instruction, the program is terminated.  
  
  
f. Write a program to subtract a number from another. List your program below (10 marks)



LOADTEMP (8)

ADD (8)

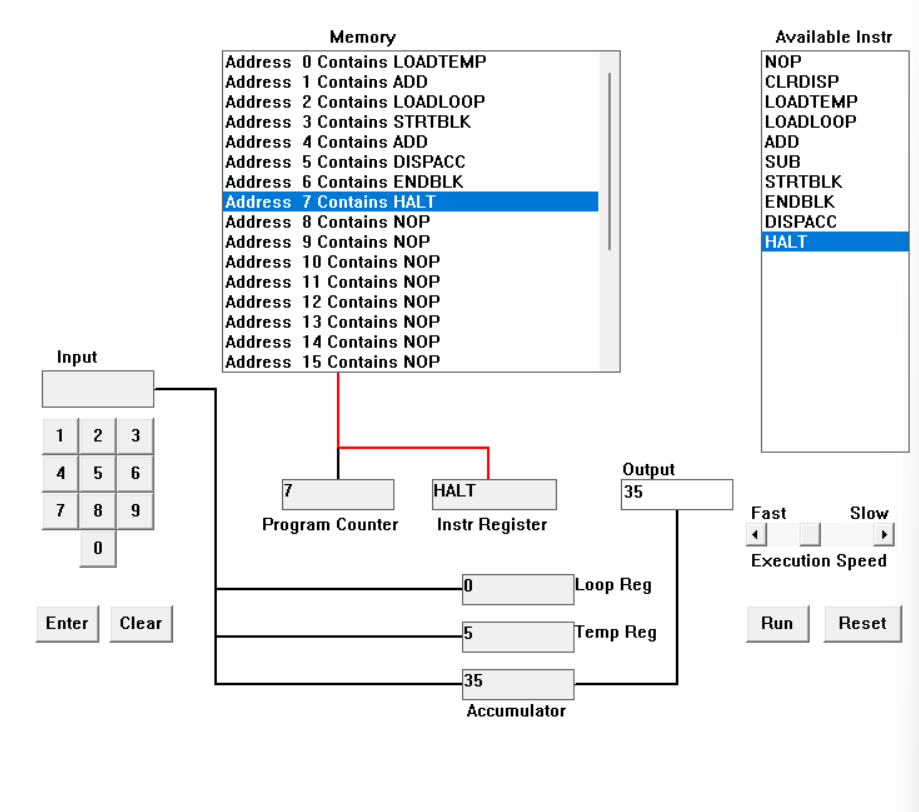
LOADTEMP (4)

SUB (4)

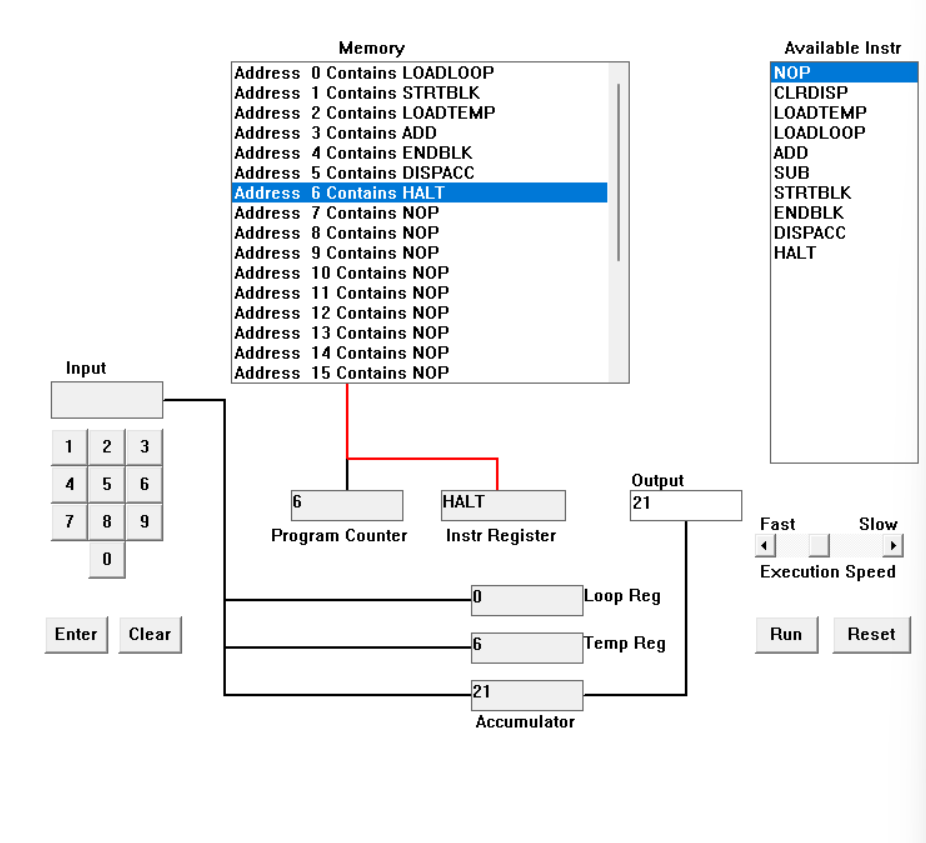
DISPACC (4)

HALT  
  
Beginning with the LOADTEMP instruction, the program loads the number 8 into a temporary register. After using the ADD instruction to multiply this number by 8, the result is added to 8 and saved in the accumulator. With the help of the LOADTEMP instruction, the program then loads the value 4 into the temporary register and subtracts it from the accumulator's 8 current value. Afterward, the accumulator stores the operation's output, 4, which is 4. The DISPACC instruction is then used by the program to show the value of 4, which is stored in the accumulator, on the screen. Using the HALT instruction, the program is terminated.

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)

LOADTEMP (5)  
ADD (5)  
LOADLOOP (6)  
STRTBLK  
ADD  
DISPACC (35)  
ENDBLK  
HALT

The first instruction, LOADTEMP, inserts the number 5 into a temporary register. The following instruction, ADD, adds the value in the temporary register to the value in the accumulator register. The LOADLOOP instruction inserts the number 6 into the loop counter register. The following instruction, STRTBLK, initiates a block of instructions that will be repeated as many times as determined by the value stored in the loop counter register. The following instruction ADD, adds the contents of the accumulator to the items in the temporary register. The value 35 and the contents of the accumulator are shown by the command DISPACC (35). The block of instructions comes to a close with the instruction ENDBLK. Finally, the HALT instruction halts the execution of the program.

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 (6)

STRTBLK

LOADTEMP (1, 2, 3, 4, 5, 6)

ADD

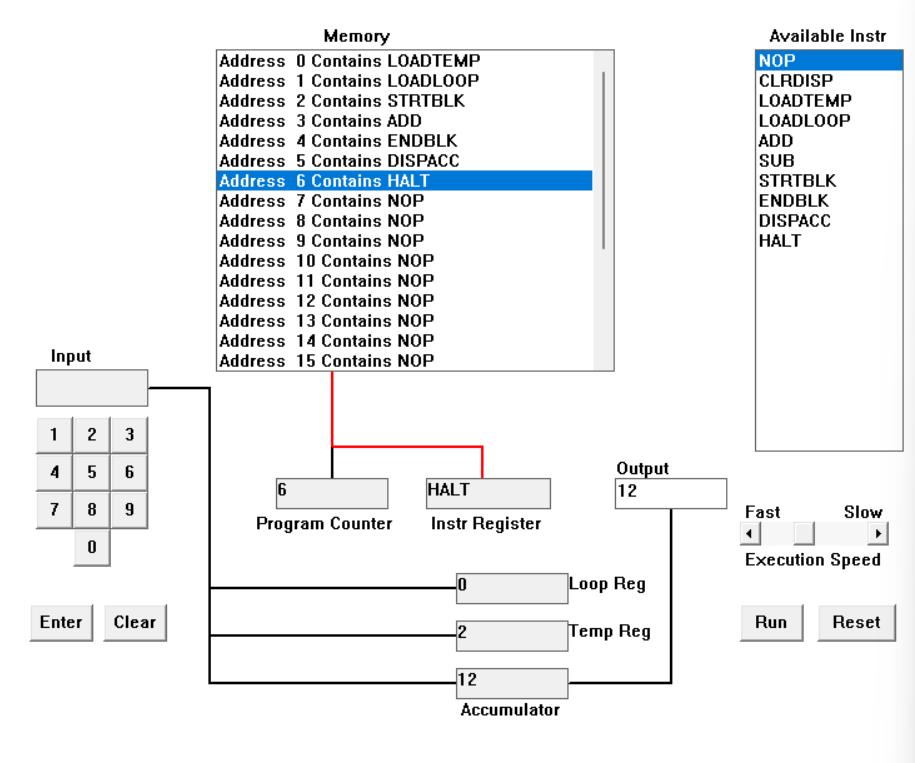
ENDBLK

DISPACC (21)

HALT

The loop counter register is loaded with the value 6 in the first instruction, LOADLOOP (6), which determines how many times the block of instructions will be repeated. A block of instructions that will be repeated six times is initiated with the command STRBLK. The numbers 1, 2, 3, 4, 5, and 6 are loaded into a temporary register in that order by the instruction LOADTEMP (1,2,3,4,5,6). The ADD instruction adds the contents of the temporary register to the contents of the accumulator register. The block of instructions inside the loop are repeated six times total after the command ENDBLK concludes the block. The instruction DISPACC (21) displays the contents of the accumulator as well as the value 21. The HALT command halts the running of the program. This program will load six different values into a temporary register and add them one by one to the accumulator register before displaying the accumulator's final value, 21, and ending the program.

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

LOADTEMP (2)

LOADLOOP (6)

STRTBLK

ADD (12)

ENDBLK

DISPACC (12)

HALT

The program loads the number 6 into the loop counter register, also known as the loop register, and the value 2 into the temporary register, commonly known as the temp register, of the CPU. Utilizing the instruction STRTBLK, it then starts a block of operations. The number of times the instruction ADD is executed in this block, accumulating the value in the accumulator register, also known as the ACC, with the value in the temp register, is determined by the value in the loop counter. The final value 12 from the accumulator is shown on the screen by the program using the instruction DISPACC (12). Using the instruction HALT, it subsequently ends the program's running. Before this, it loads the value 2 into the temporary register and the value 6 into the loop register and begins a block of operations. This block includes repeatedly adding the value from the temporary register to the accumulator, as controlled by the value in the loop counter. Finally, it displays the final value of 12 on the screen and ends execution.