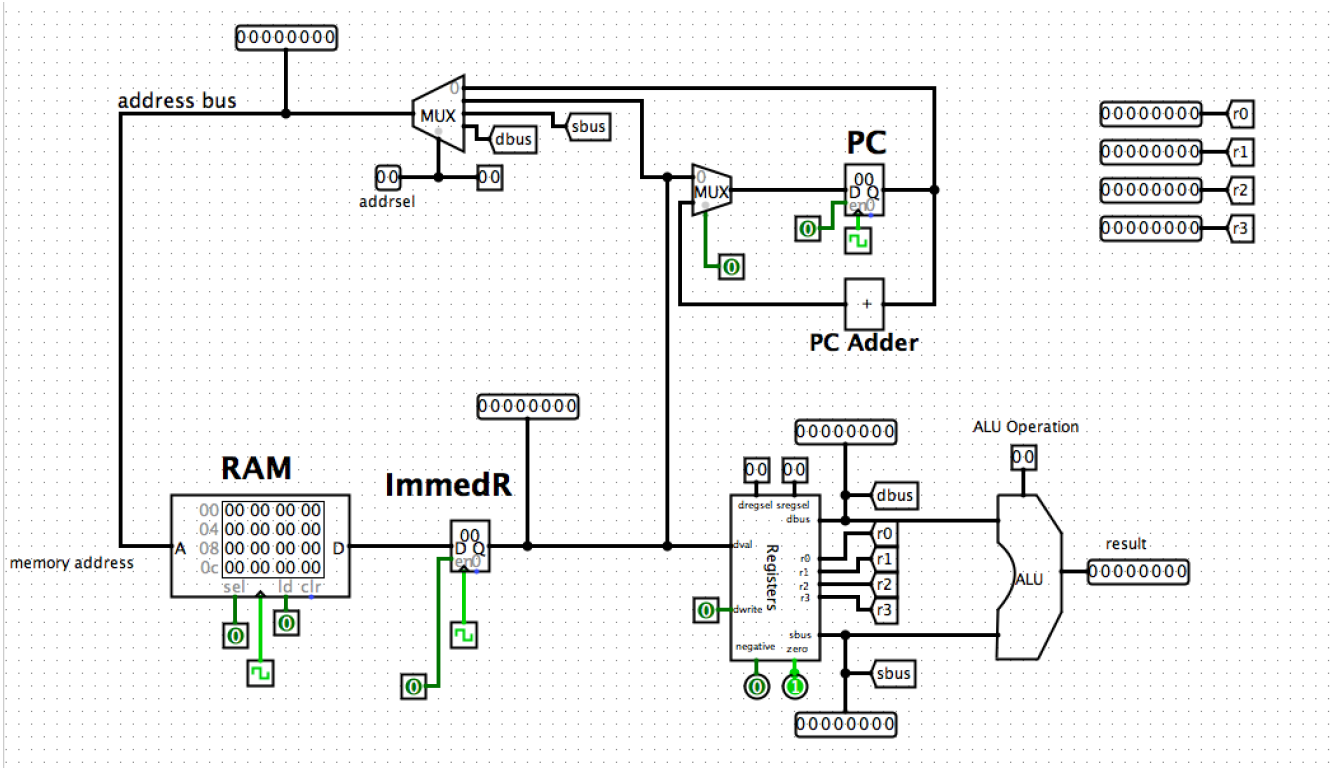
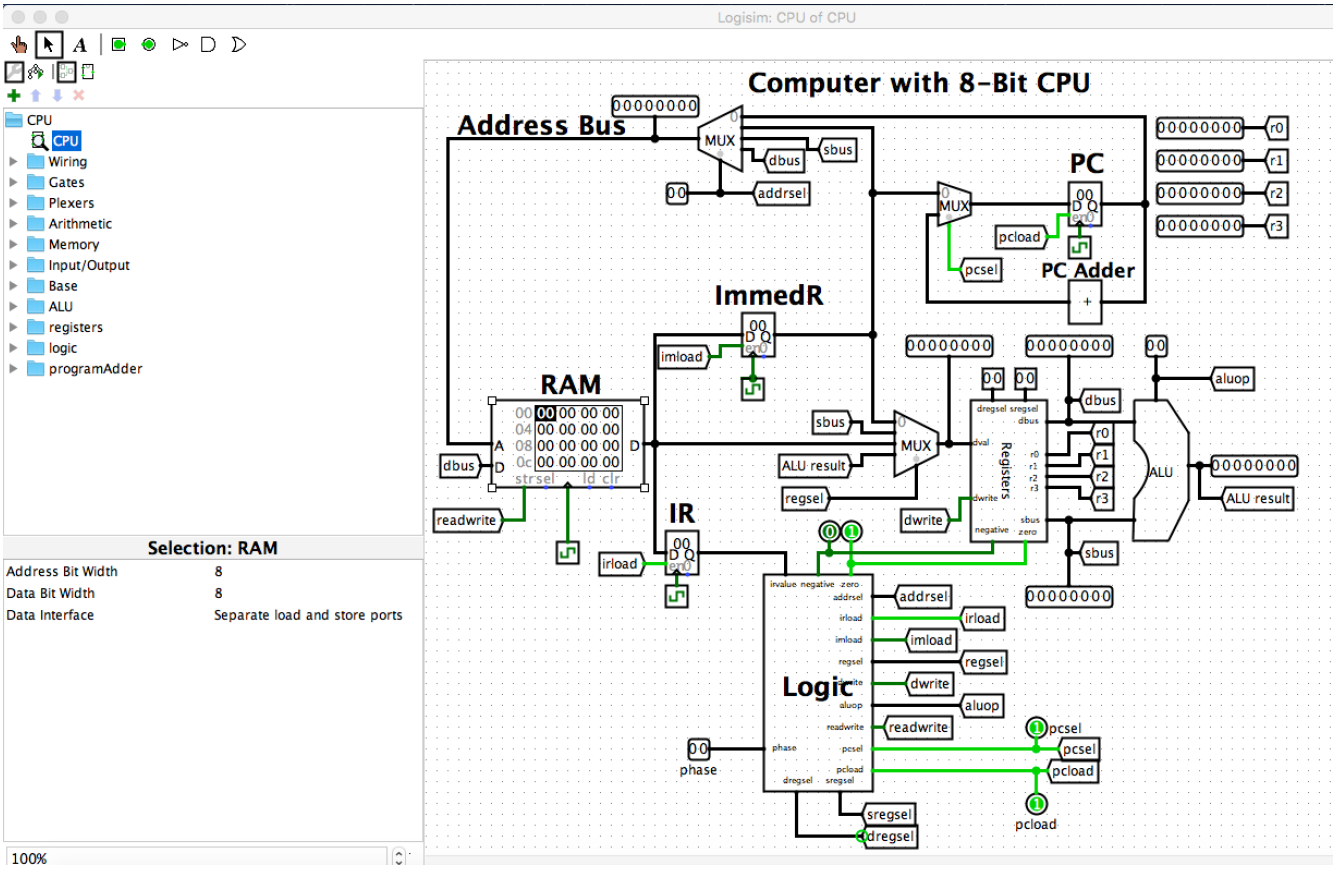
ASDV 1205
Lab, CPU complete

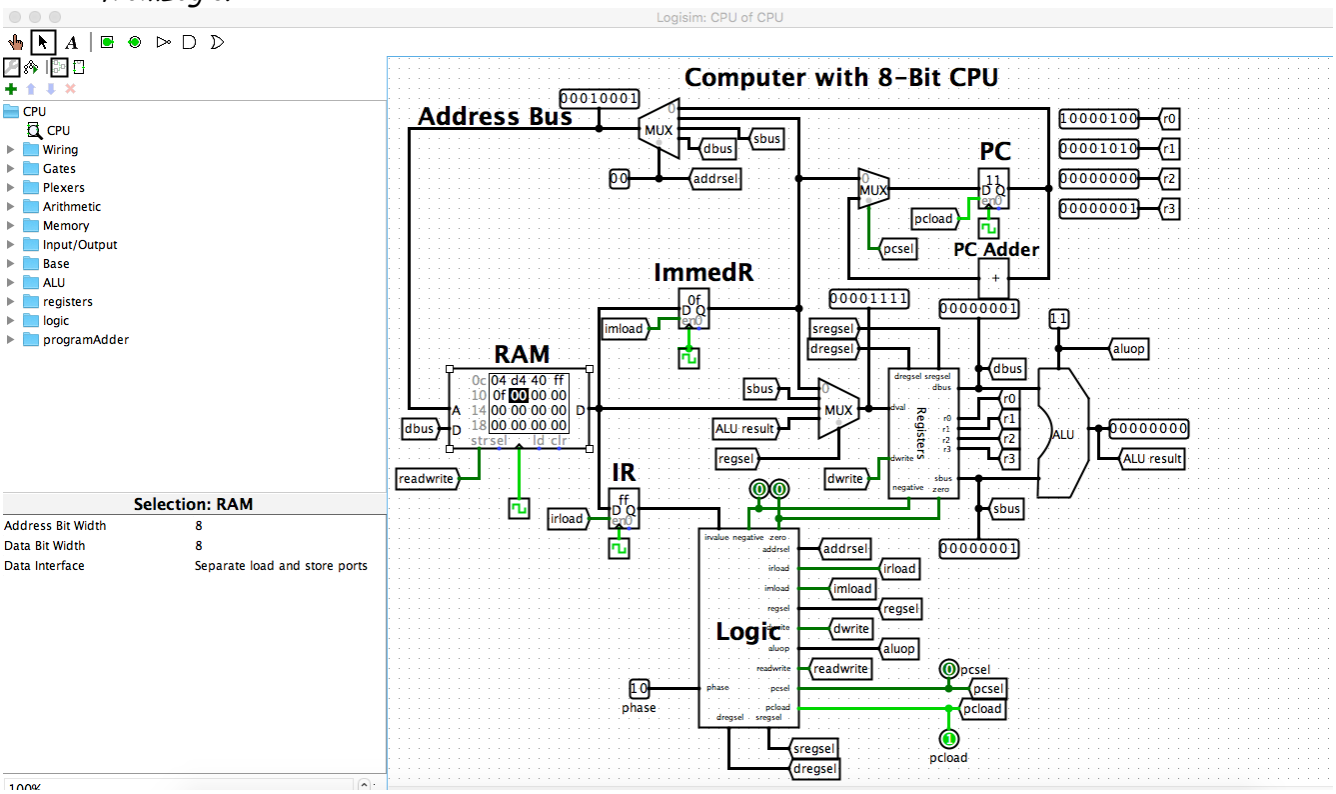
1. Open your last circuit that does not contain the LOGIC circuit. Your last circuit shoul have all but LOGIC chip and the IR register as shown below:



2. Modify its RAM that has 2 separate ports for read (load) and write(store) by clicking the RAMs settings. Connect the store-port to dbus.
3. Insert the IR (Instruction Register) and the LOGIC chip. Connect the pieces to create the working computer shown below:



4. Right click on RAM and select Load Image. Load the posted executable program which is in the file called *cpumemAdd.txt* Enable ticks for the clock to give pulses and run the program. When your program enters the infinite loop in the last instruction r0 shows 4 (100), r1 shows 10 (1010) as shown below. *In Registers chip, replace the inputs sregsel, dregsel with tunnels that come fromLogic.*



5. Disable the ticks. Click Simulate, Reset Simulation. Right clikc on RAM , selet Clear Contents. Load into RAM the program again. Run the program one instruction at a time slowly. From Simulate select *tick once*. By ticking once, you can execute the program in an observable manner.

The assembly code for the program.

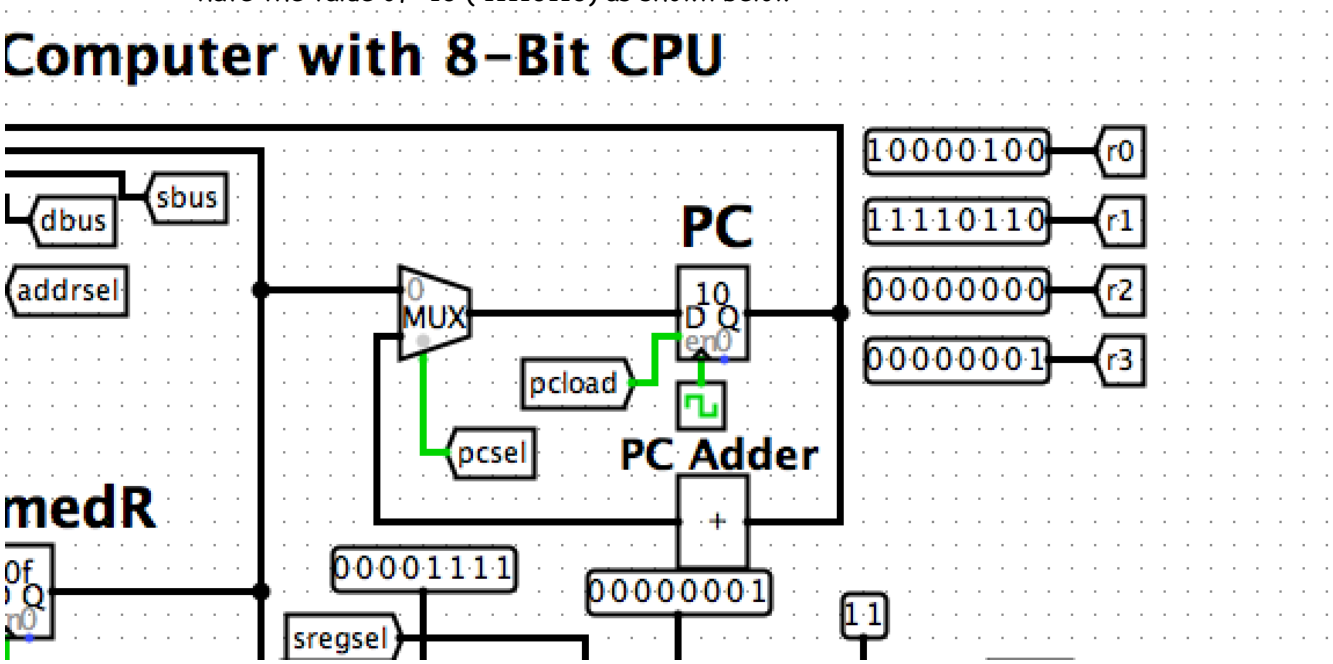
```
 LI   R1,0x00          # Set running sum to zero
      LI   R0,0x80     # Start at beginning of list
loop: LW   R2, (R0)    # Get the next number
      JEQ  R2, end     # Exit loop if number == 0
      ADD  R1, R2      # Add number to running sum
      LI   R3, 0x01    # Put 1 into R3, so we can do
      ADD  R0, R3      # R0++
      JMP  loop        # Loop back
end:  SW   R1, 0x40    # Store result at address 0x40
inf:  JMP  inf         # Infinite loop
```

LI R1,0x00	e4 00
LI R0,0x80	e0 80
LW R2, (R0)	48
JEQ R2, end	88 0d
ADD R1, R2	26
LI R3, 0x01	ec 01
ADD R0, R3	23
JMP loop	ff 04
SW R1, 0x40	d4 40
JMP inf	ff 0f

•

6.Upload the **r1_1010.jpg** of your computer with the program running in final infinite loop. If you don't have a 1010 in R1 at the end then your computer is not working properly, fix it.

7. Open the program *cpumemAdd.txt* using a PLAIN text edit, not Word.
8. Modify the program to do subtraction of all numbers not addition. Save the program as a text file *cpumemSubtraction.txt.*
9. Load the new program and run it. The final results in registers are shown below. R1 should have the value of -10 (11110110) as shown below:



10. **Upload the file minus10cpu.jpg** that shows the whole computer of course not just r1, with r1 being -10.

EXTRA Credit : 20 points

Creates a program that reads 5 numbers exactly using a counter loop. Your program should find the minimum of the 10 numbers.

These 5 numbers are 20, 3, -1, 33 1.

Your program should display the minimum in register R3 and the counter variable in register R0.