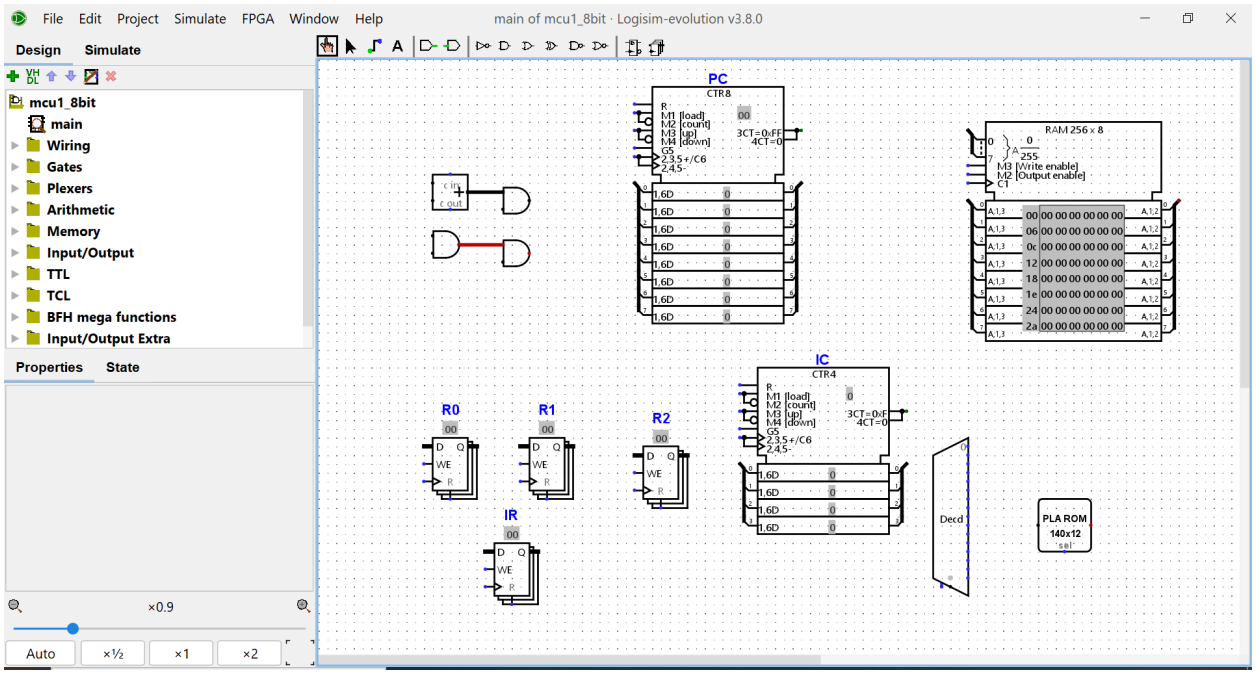

Figure: Barebone blocks for 8-bit microprocessor

Create a 8-bit microprocessor with following features

- Separate address and data bus, each 8 bit
- 3 core registers (R0, R1, R2), two for inputs into ALU, one for output of ALU
- ALU should have addition and logical AND operations
- Use Logisim evolution 3.8.0 version

Minimum capabilities:

Enable signals:

E0 = RAM OE
E1 = RAM IE
E2 = Clock EN for PC
E3 = Clock EN for R0
E4 = Clock EN for R1
E5 = Clock EN for R2
E6 = Clock EN for IR
E7 = Clock EN for RAM
E8 = Load/Count for PC
E9 = Clock EN for IC
E10 = Reset for IC
E11 = Enable for ADD
E12 = Enable for AND

IR = Instruction Register; PC = Program Counter; IC = Instruction Counter

Instructions (choose suitable 8-bit opcodes) =

JUMP to some address
Read from RAM into R0 (value is just after opcode)
Read from RAM into R1 (value is just after opcode)
Write to RAM from R2, arbitrary address
ADD R0,R1 and store in R2
AND R0,R1 and store in R2
Copy R2 into R1
Copy R2 into R0

—---

Write a sample program like below and show the execution step by step:

Addr: Instruction

—---

0x00: MOV R0, val1 (val1 is in RAM just after opcode)
0x02: MOV R1, val2 (val2 is in RAM just after opcode)
0x04: ADD R0, R1
0x05: MOV @addr1, R2 (@addr1 is pointing to val1 address)
0x07: COPY R0, R2
0x08: MOV R1, val3 (val3 is in RAM just after opcode)
0x0A: AND R0, R1
0x0B: MOV @addr2, R2 (@addr2 is pointing to val2 address)
0x0D: JUMP 0x00 (jump address is 0x00)

NOTE:

- Assignment 2 has to be demonstrated in live class
- We will ask viva questions
- Team of 3 members can be formed among students for attempting this (we will have 11 teams total)
- April 16 and 18 are the demo days