Testing Procedures/Results

Testing all 11 operations shows all components (the PC, RAM Block, IR, State Machine, Decoder, Register File, and ALU) are working properly since they are all connected to each, they work together to create the 4 bit processor.

1.

MOV A, IMM Opcode: 0001, Write Register: 0, WR: Enabled, IL : Enabled, ALU: No Operation

Fetch:

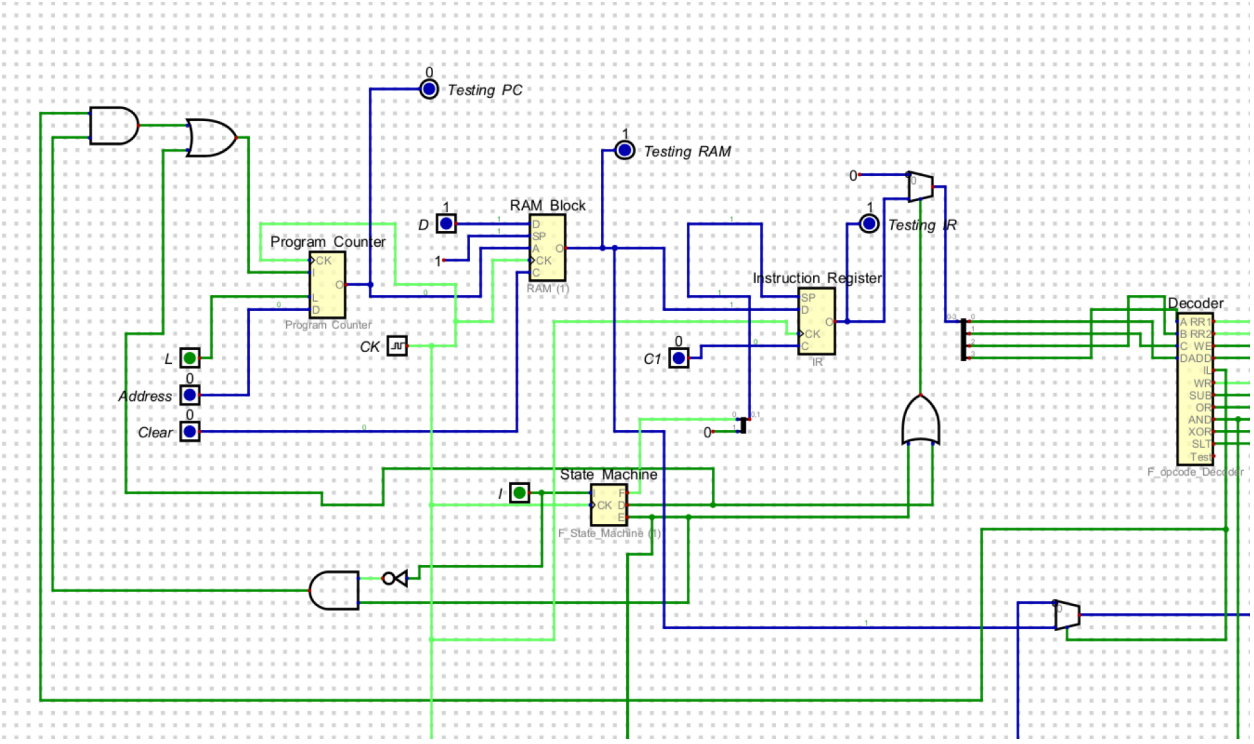
Starting PC - Address: 0

RAM Block Data: 1 (Opcode for Immediate Load A) - Keep I off and write data 1 in the RAM Block.

2 Clock Cycles: 1 for writing to RAM block, and 1 for loading value to IR - Go through 2 clock cycles to load it into the IR.

I for Finite State Machine = 0 to stay in Fetch cycle - Keep off to load into IR so we can stay in the fetch cycle

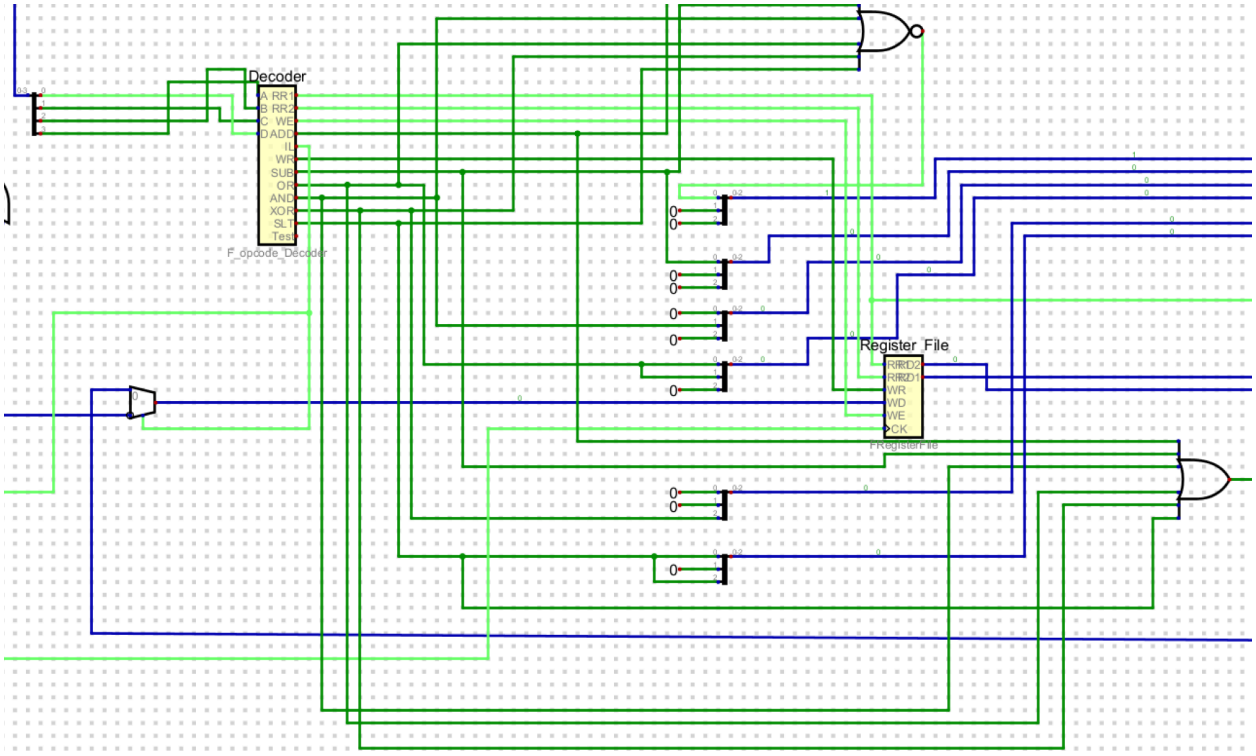
Then turn I = 1, go through 1 clock cycle to switch to decode cycle



Decode:

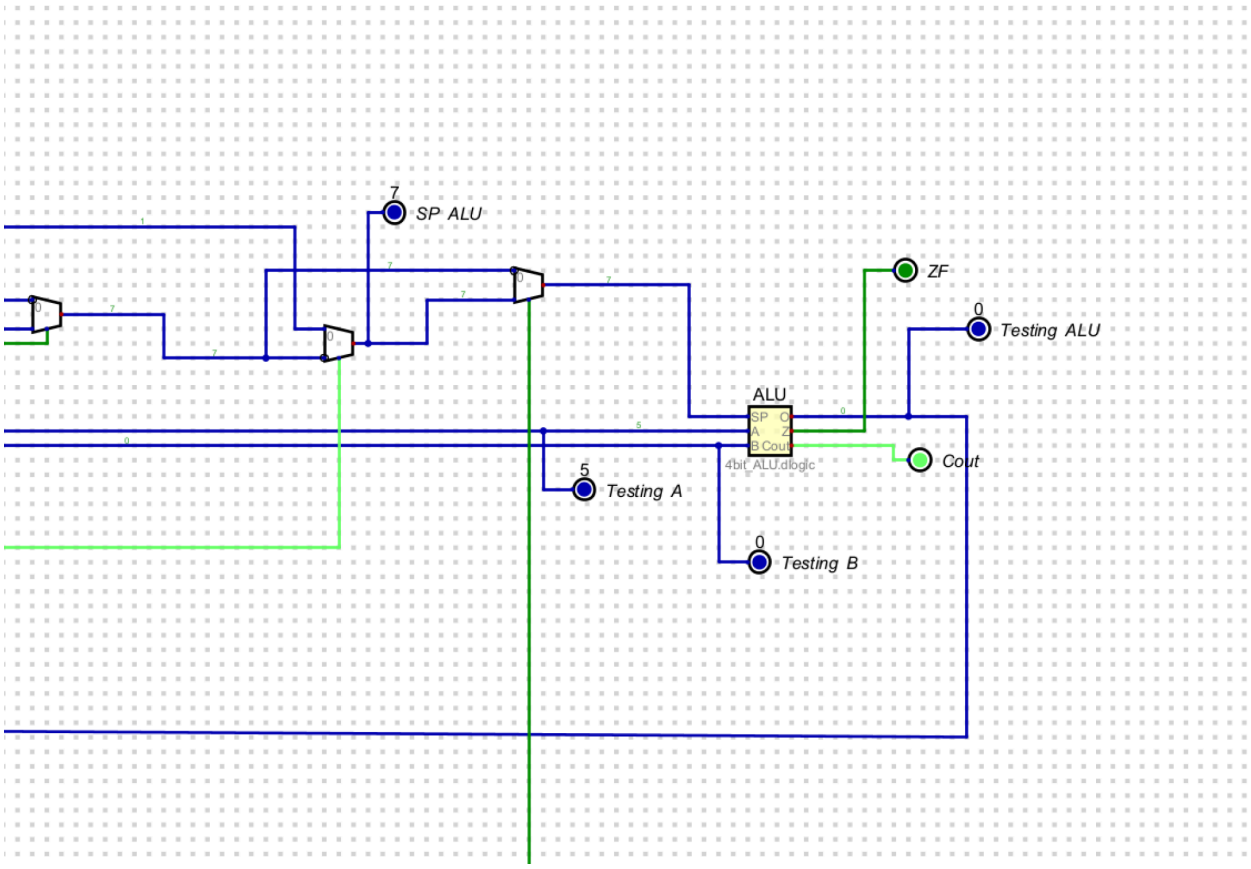
Now PC has incremented by 1 now write data to RAM Block to load immediate value to A, I will do 5

Keep I = 1, and do another clock cycle to enter execute cycle



Execute:

Now turn off I (I = 0) go through 1 clock cycle, then turn back I (I = 1) to increment the PC again since we are immediate loading, now go through 1 clock cycle to go back to Fetch.



2.

MOV B, IMM Opcode: 0010, Write Register: 1, WR: Enabled, IL : Enabled, ALU: No Operation

Fetch:

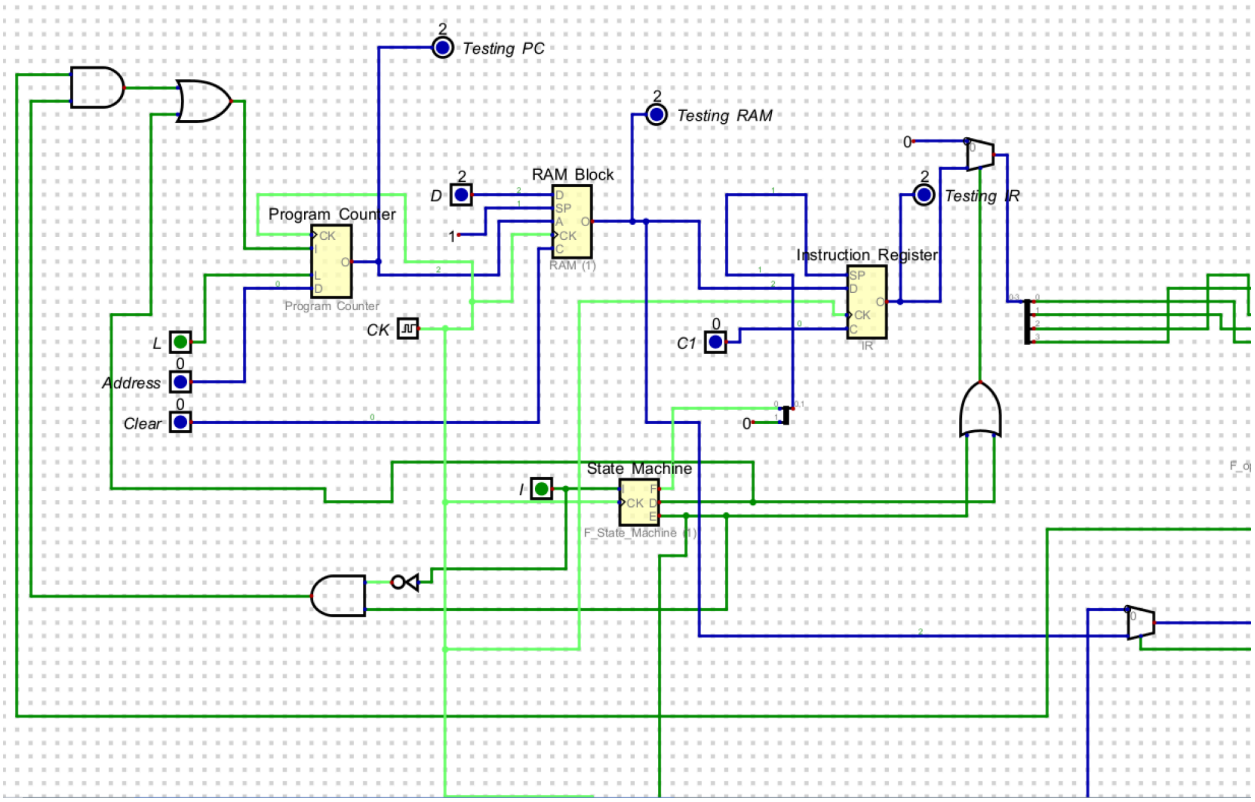
Starting PC - Address: 2

RAM Block Data: 2 (Opcode for Immediate Load B) - Now turn off I and write data 2 in the RAM Block.

2 Clock Cycles: 1 for writing to RAM block, and 1 for loading value to IR - Go through 2 clock cycles to load it into the IR.

I for Finite State Machine = 0 to stay in Fetch cycle - Keep off to load into IR so we can stay in the fetch cycle

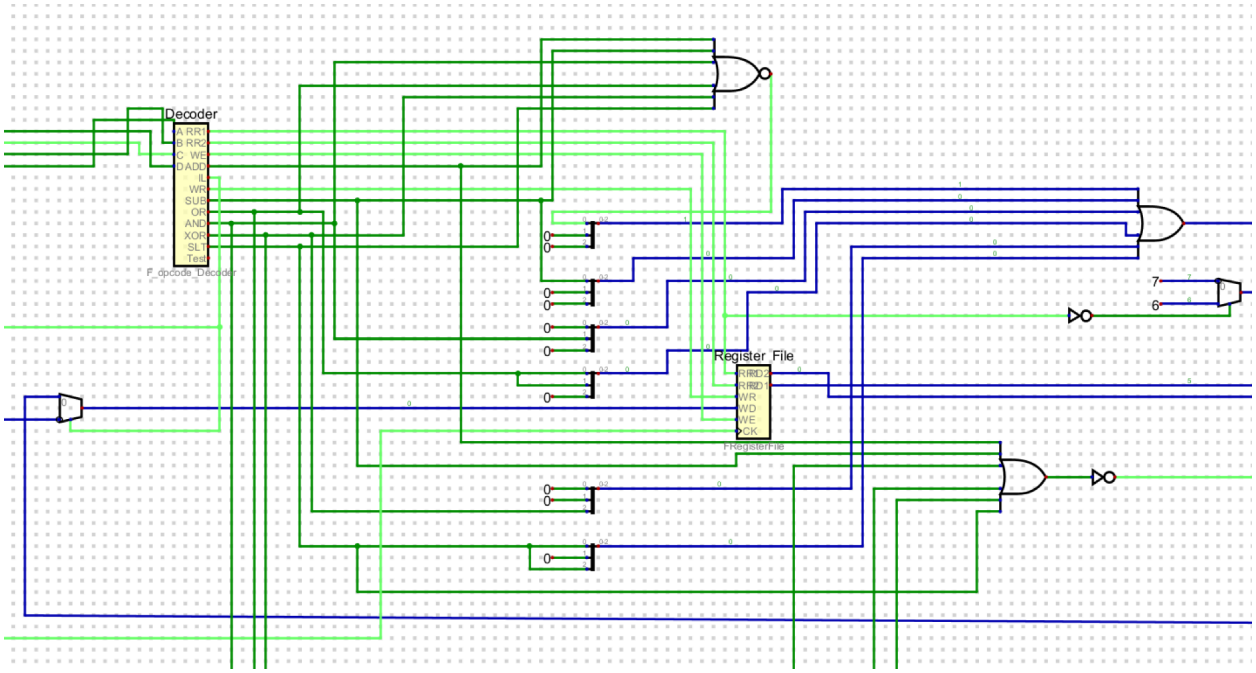
Then turn I = 1, go through 1 clock cycle to switch to decode cycle



Decode:

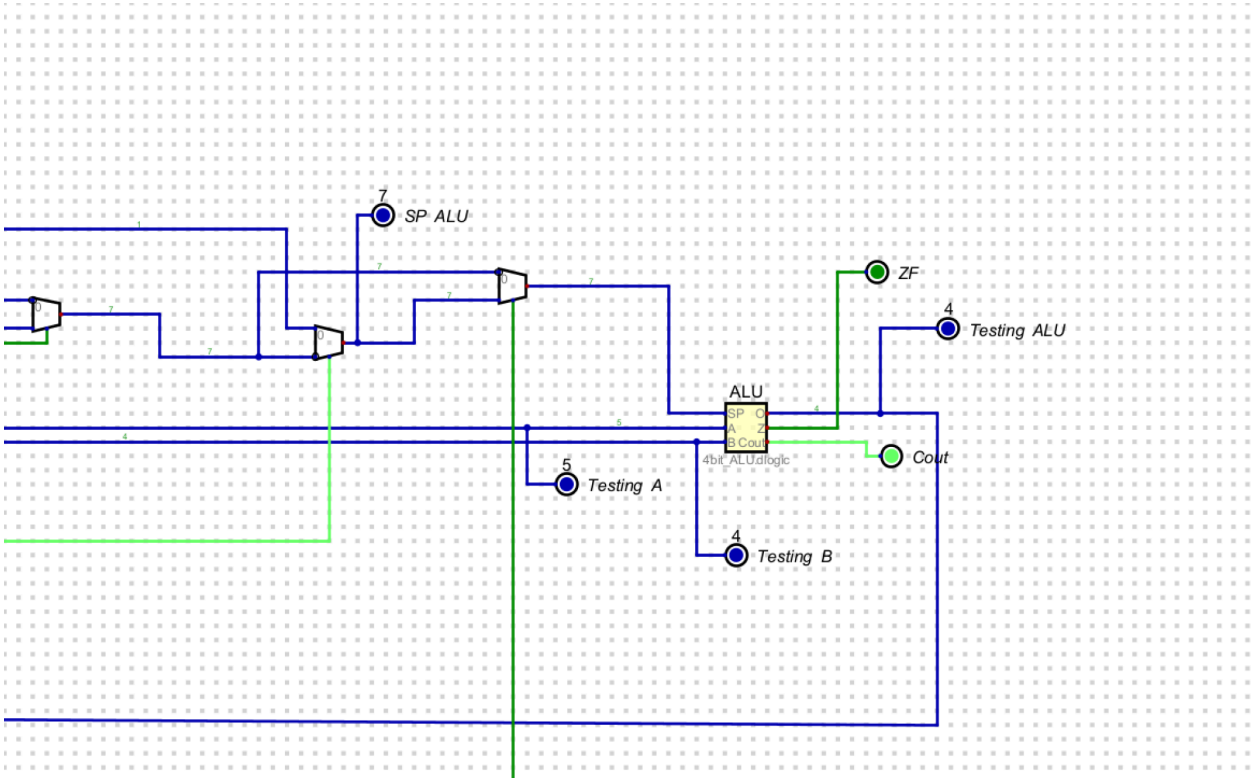
Now PC has incremented by 1 now write data to RAM Block to load immediate value to B, I will do 4

Keep I = 1, and do another clock cycle to enter execute cycle



Execute:

Now turn off I (I = 0) go through 1 clock cycle, then turn back I (I = 1) to increment the PC again since we are immediate loading, now go through 1 clock cycle to go back to Fetch.



3.

ADD Opcode: 1001, Read Register 1: 0, Read Register 2: 1, Write Register: 0, Write Enable: Enabled, Immediate Load: Disabled, ALU Operation: ADD

Fetch:

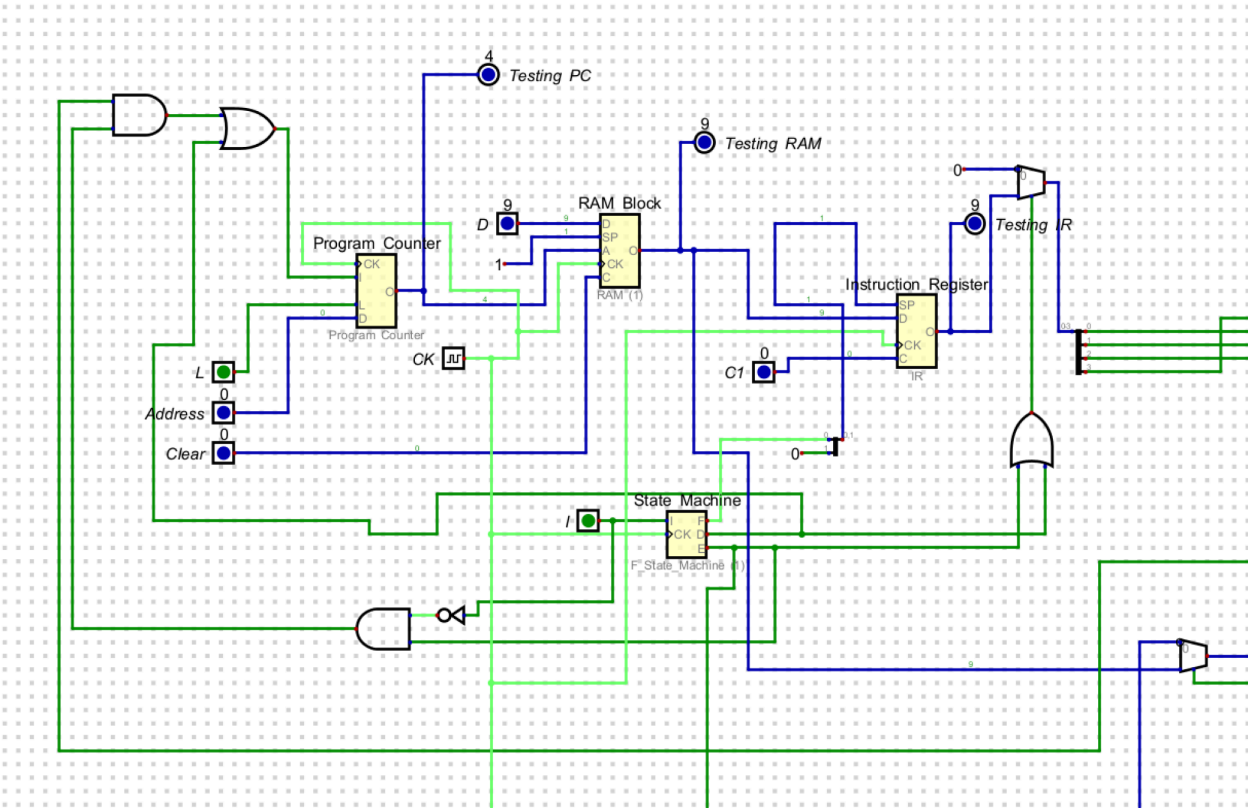
Starting PC - Address: 4

RAM Block Data: 9 (Opcode for ADD) - Now turn off I and write data 9 in the RAM Block

2 Clock Cycles: 1 for writing to RAM block, and 1 for loading value to IR - Go through 2 clock cycles to load it into the IR

I for Finite State Machine = 0 to stay in Fetch cycle - Keep off to load into IR so we can stay in the fetch cycle

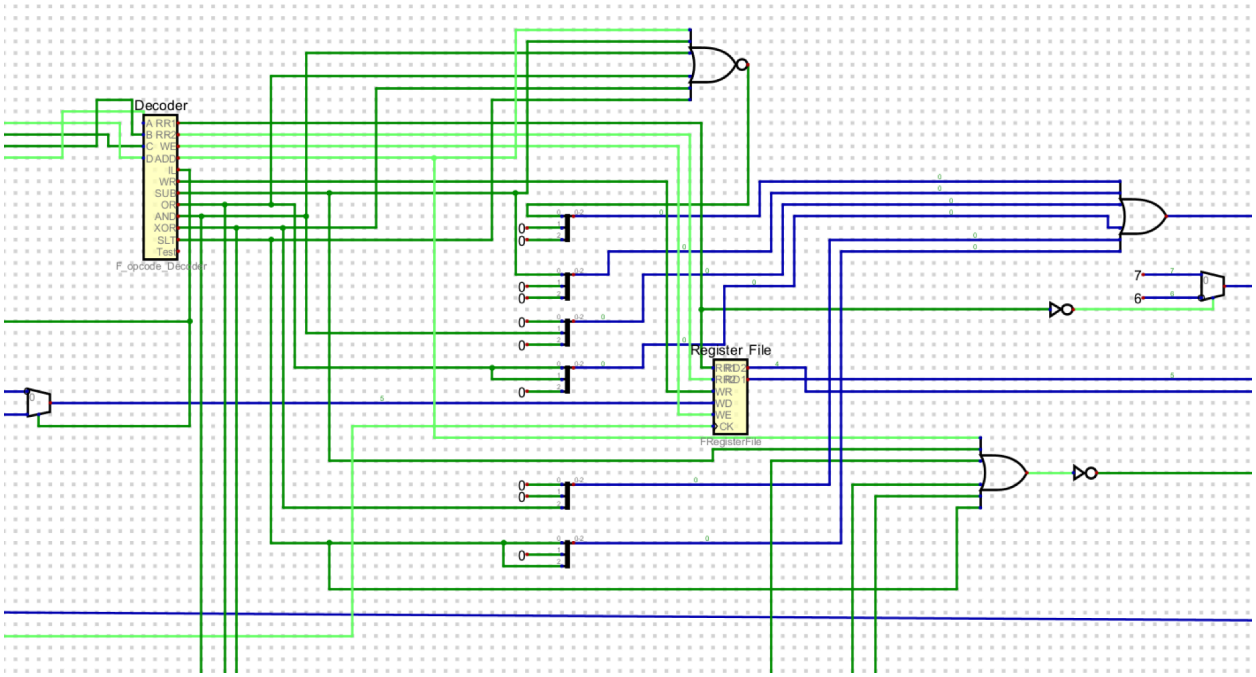
Then turn I = 1, go through 1 clock cycle to switch to decode cycle



Decode:

Now PC has incremented by 1, now we are not writing any immediate values

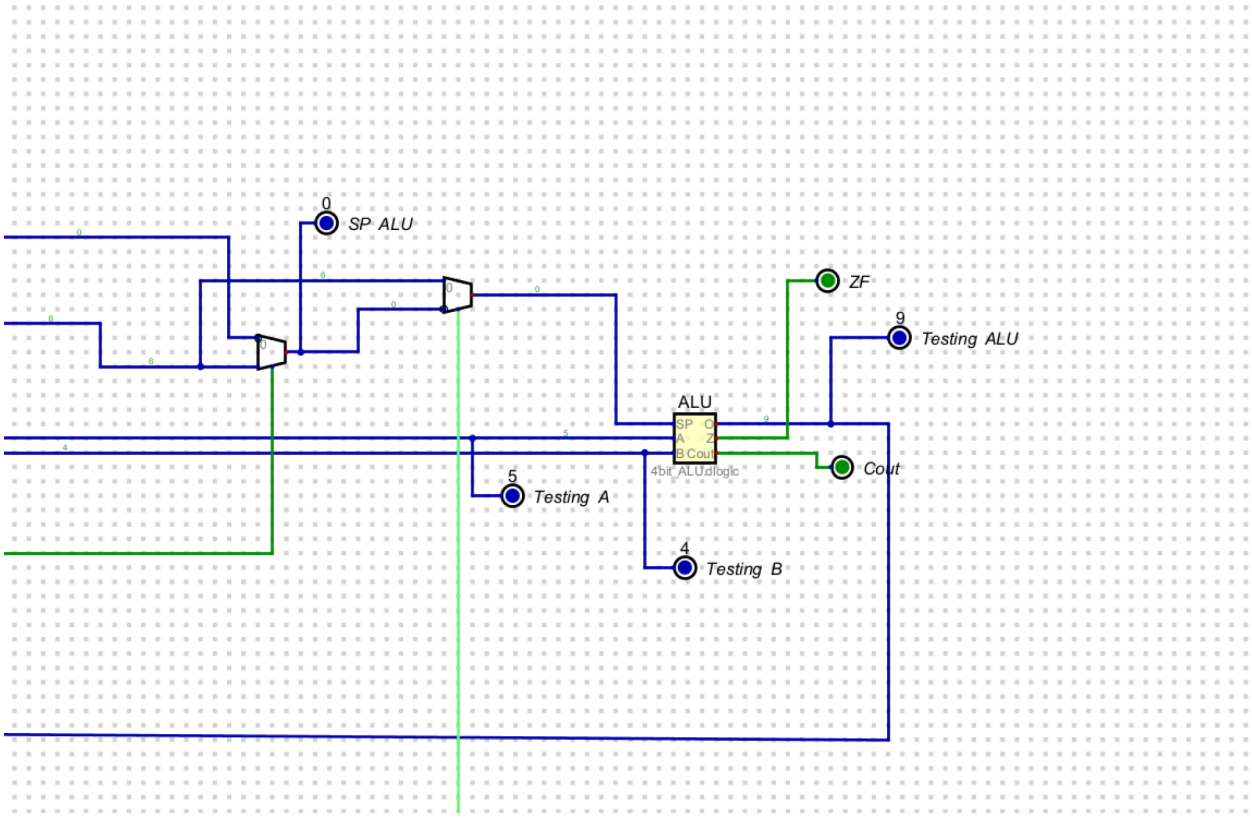
Keep I = 1, and do another clock cycle to enter execute cycle



Execute:

The ALU output is now 9 (5 + 4 = 9)

now go through the 1 clock cycle to go back to Fetch.



4.

SUB Opcode: 1010, Read Register 1: 0, Read Register 2: 1, Write Register: 0, Write Enable: Enabled, Immediate Load: Disabled, ALU Operation: SUB

Fetch:

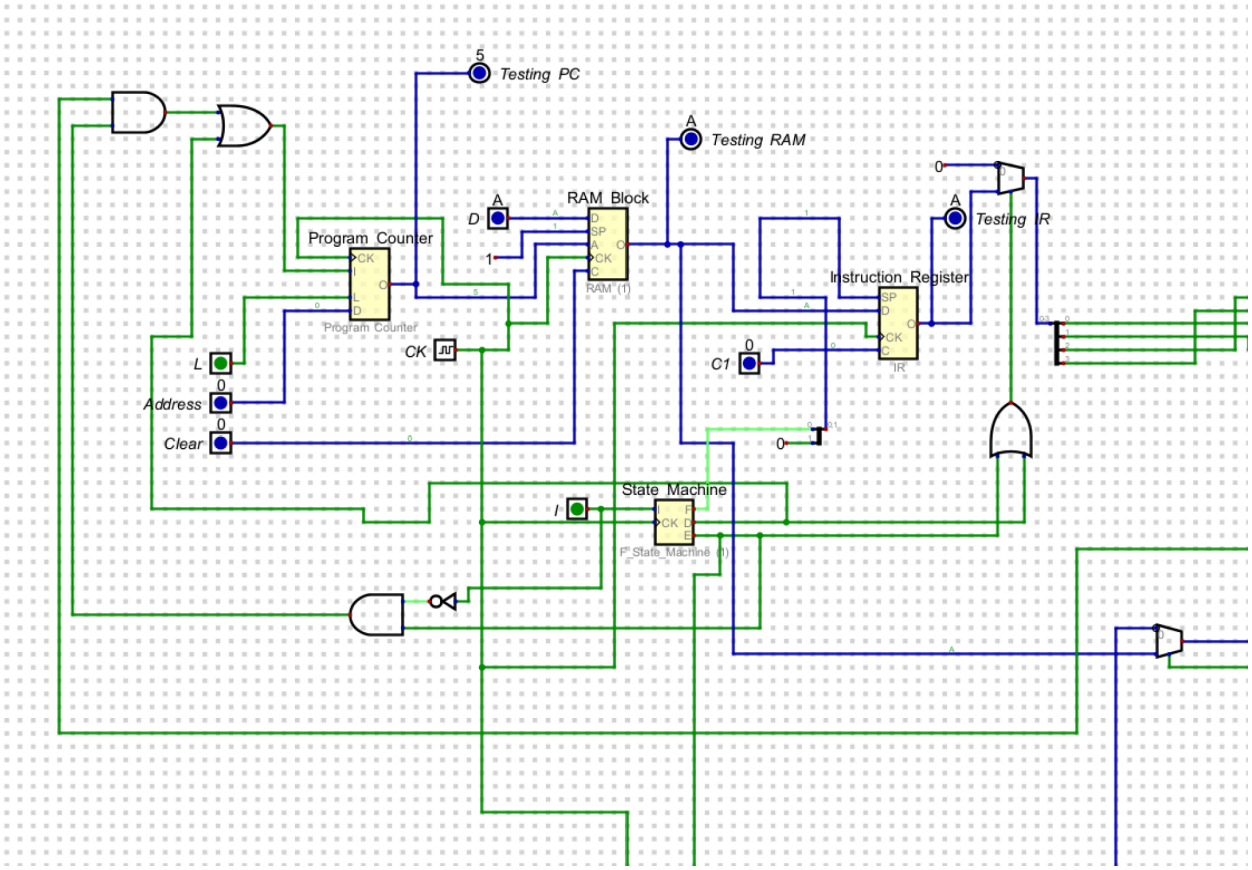
Starting PC - Address: 5

RAM Block Data: 10 (Opcode for SUB) - Now turn off I and write data 10 in the RAM Block

2 Clock Cycles: 1 for writing to RAM block, and 1 for loading value to IR - Go through 2 clock cycles to load it into the IR

I for Finite State Machine = 0 to stay in Fetch cycle - Keep off to load into IR so we can stay in the fetch cycle

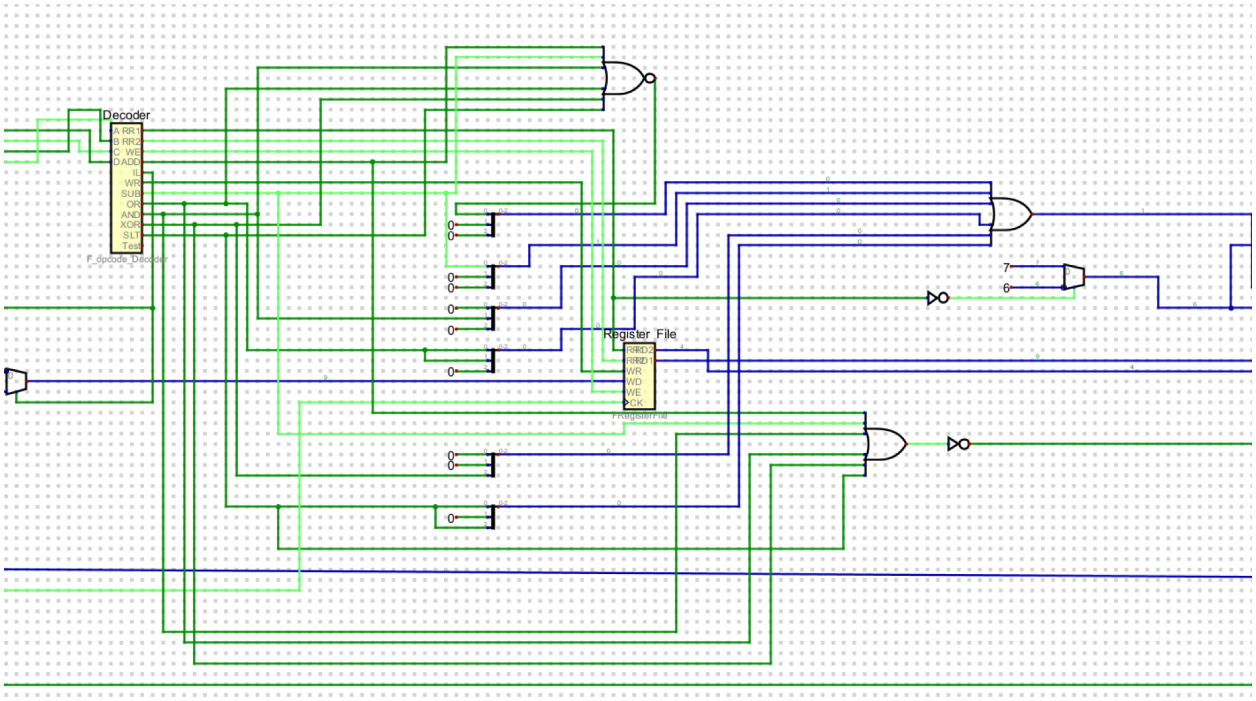
Then turn I = 1, go through 1 clock cycle to switch to decode cycle



Decode:

Now PC has incremented by 1, now we are not writing any immediate values

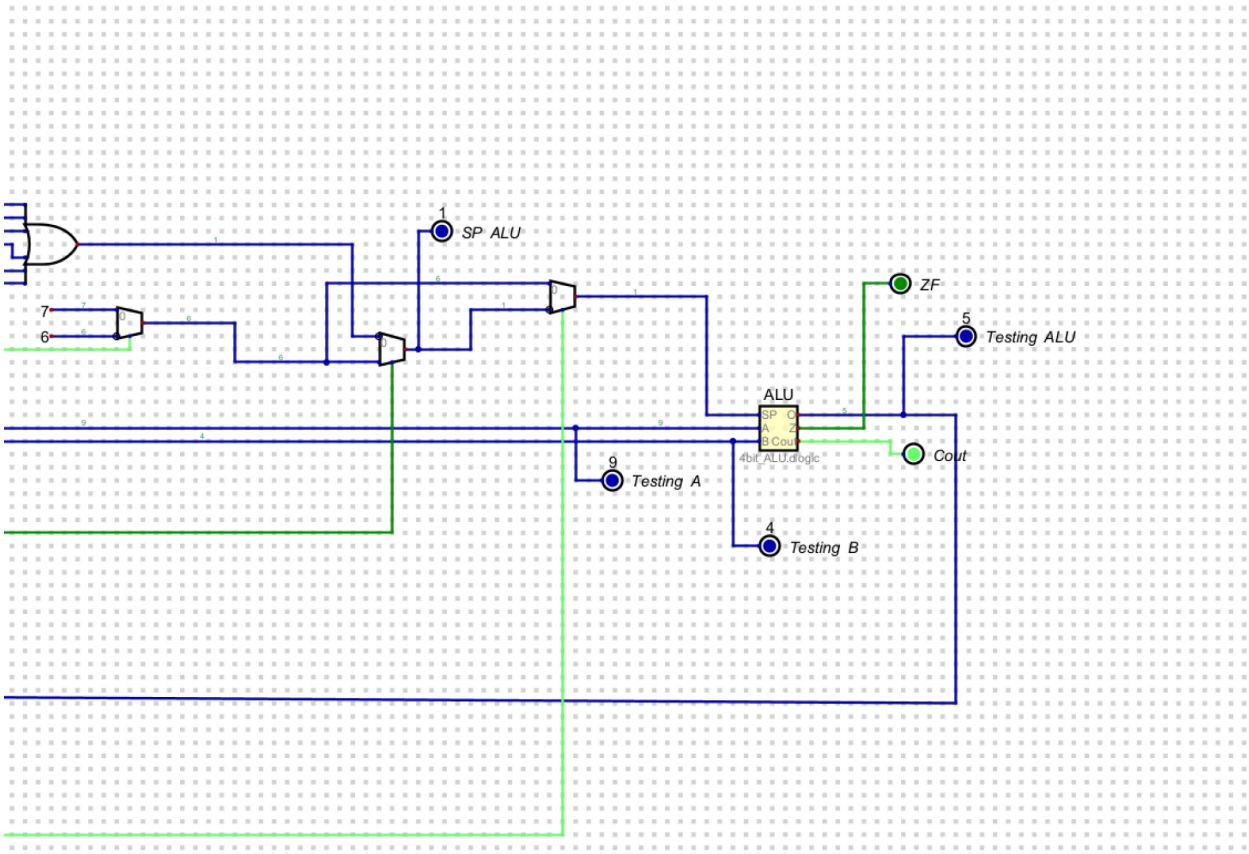
Keep I = 1, and do another clock cycle to enter execute cycle



Execute:

The ALU output is now 5 (9 - 4 = 5)

now go through the 1 clock cycle to go back to Fetch.



5.

AND Opcode: 1011, Read Register 1: 0, Read Register 2: 1, Write Register: 0, Write Enable: Enabled, Immediate Load: Disabled, ALU Operation: AND

Fetch:

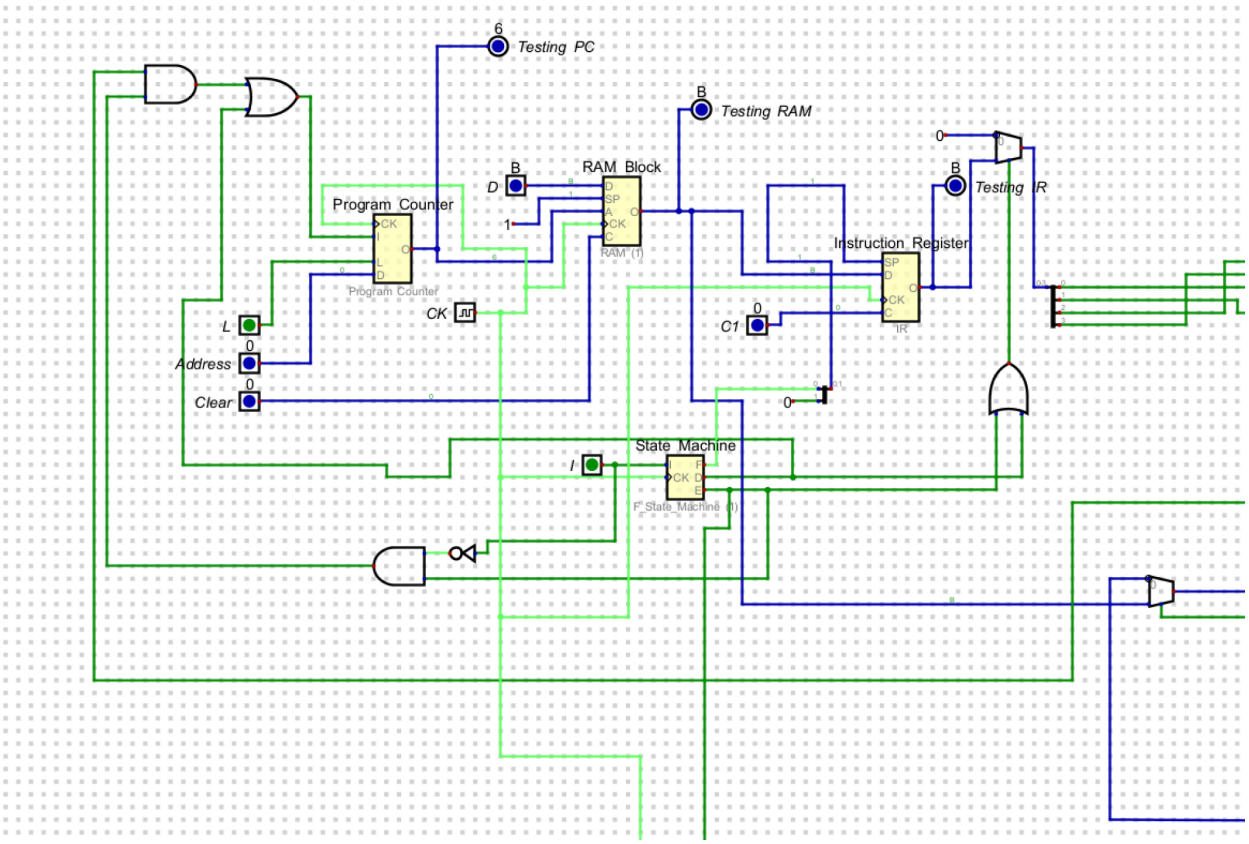
Starting PC - Address: 6

RAM Block Data: 11 (Opcode for AND) - Now turn off I and write data 11 in the RAM Block

2 Clock Cycles: 1 for writing to RAM block, and 1 for loading value to IR - Go through 2 clock cycles to load it into the IR

I for Finite State Machine = 0 to stay in Fetch cycle - Keep off to load into IR so we can stay in the fetch cycle

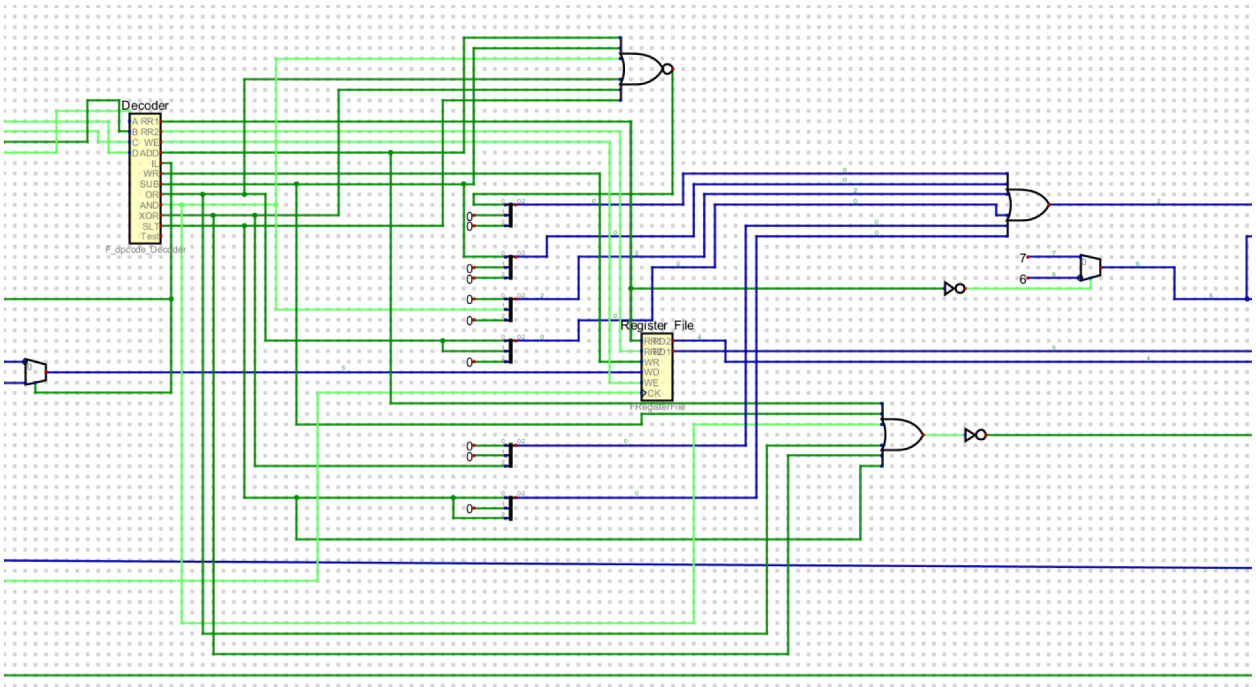
Then turn I = 1, go through 1 clock cycle to switch to decode cycle



Decode:

Now PC has incremented by 1, now we are not writing any immediate values

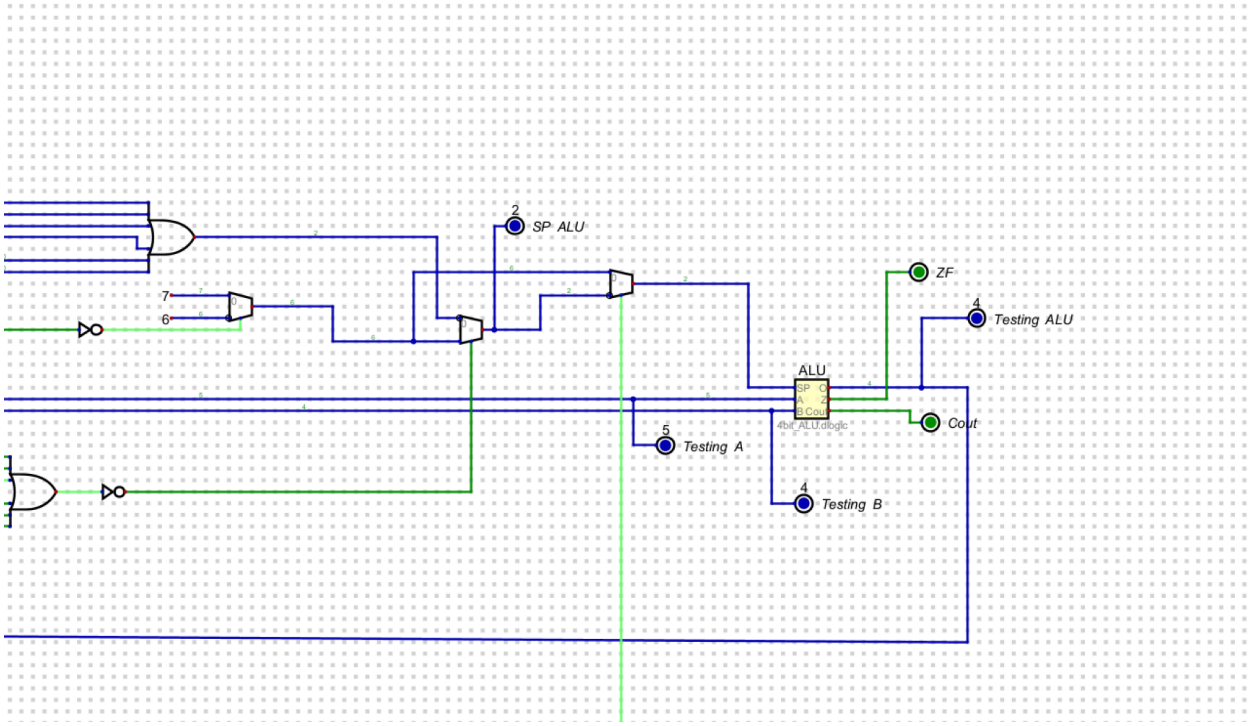
Keep I = 1, and do another clock cycle to enter execute cycle



Execute:

The ALU output is now 4 (5 AND 4 = 4)

now go through the 1 clock cycle to go back to Fetch.



6.

OR Opcode: 1100, Read Register 1: 0, Read Register 2: 1, Write Register: 0, Write Enable: Enabled, Immediate Load: Disabled, ALU Operation: OR

Fetch:

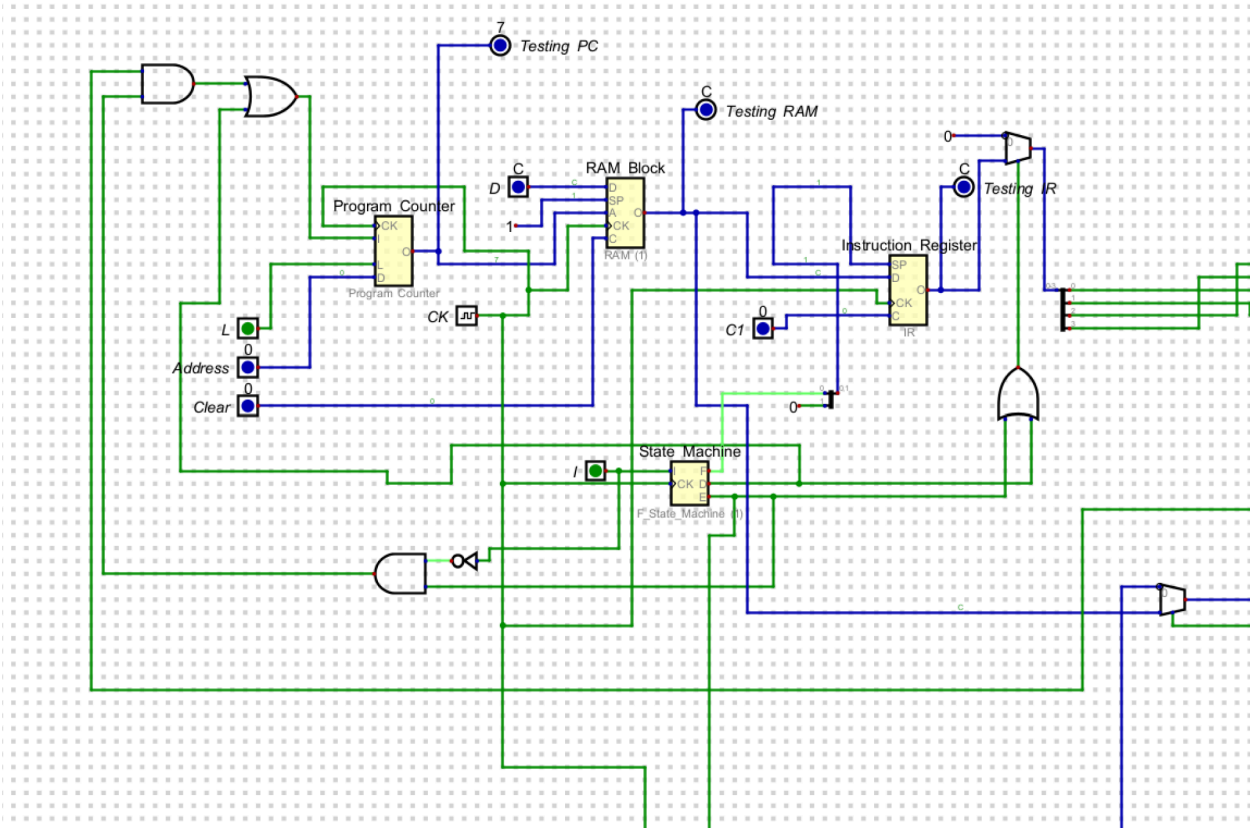
Starting PC - Address: 7

RAM Block Data: 12 (Opcode for OR) - Now turn off I and write data 12 in the RAM Block

2 Clock Cycles: 1 for writing to RAM block, and 1 for loading value to IR - Go through 2 clock cycles to load it into the IR

I for Finite State Machine = 0 to stay in Fetch cycle - Keep off to load into IR so we can stay in the fetch cycle

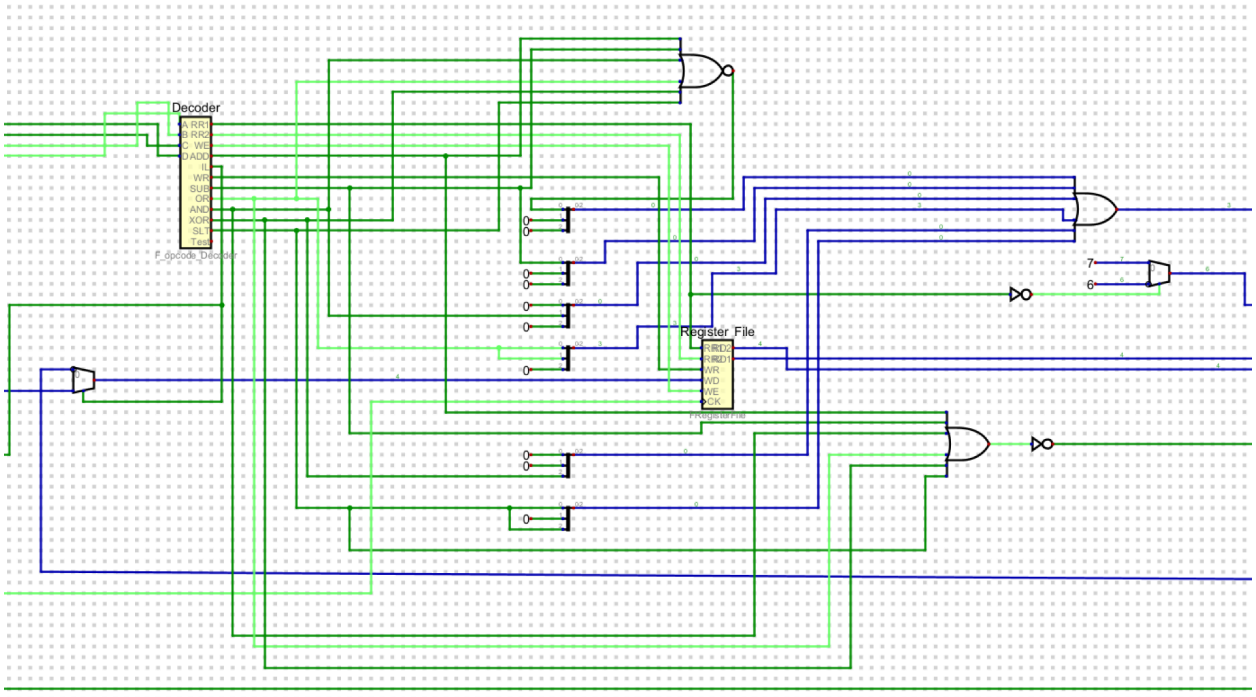
Then turn I = 1, go through 1 clock cycle to switch to decode cycle



Decode:

Now PC has incremented by 1, now we are not writing any immediate values

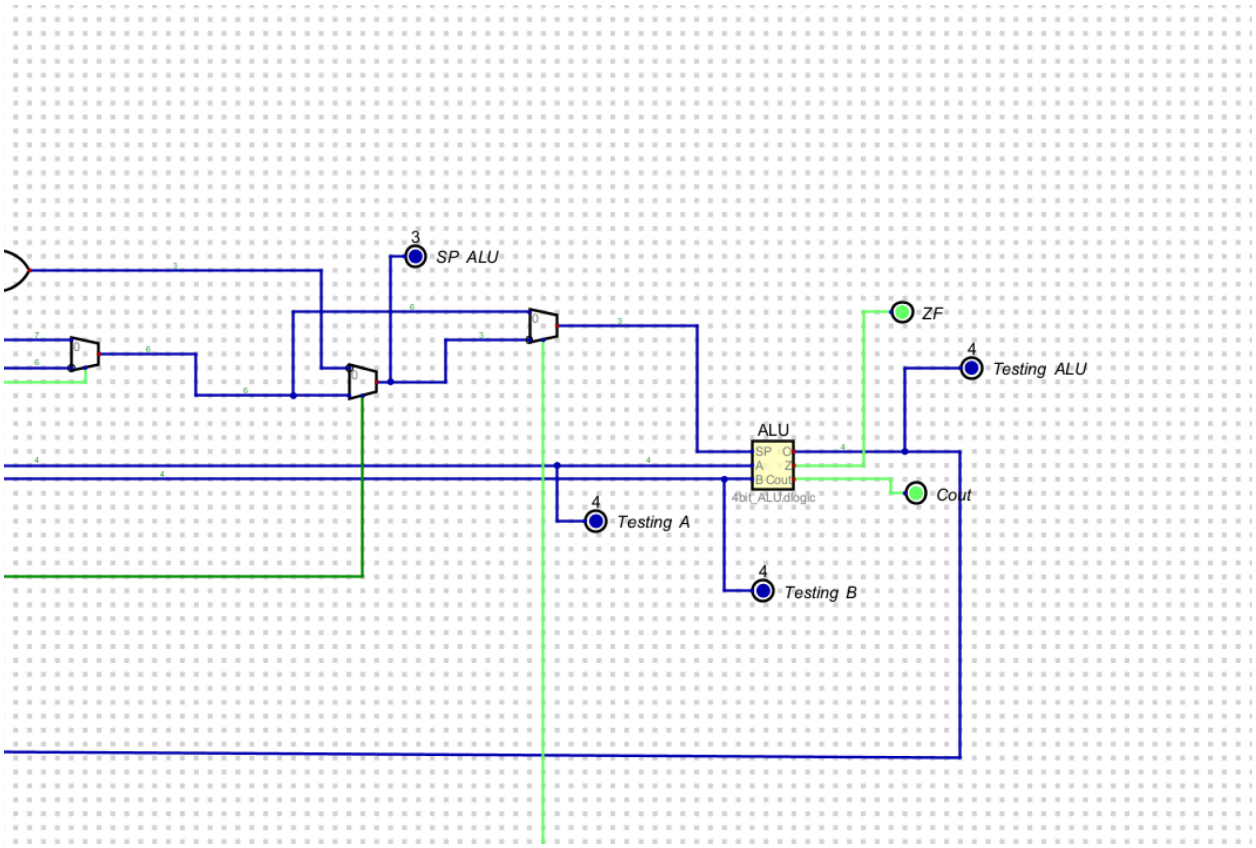
Keep I = 1, and do another clock cycle to enter execute cycle



Execute:

The ALU output is now 4 (4 OR 4 = 4)

now go through the 1 clock cycle to go back to Fetch.



7.

XOR Opcode: 1101, Read Register 1: 0, Read Register 2: 1, Write Register: 0, Write Enable: Enabled, Immediate Load: Disabled, ALU Operation: XOR

Fetch:

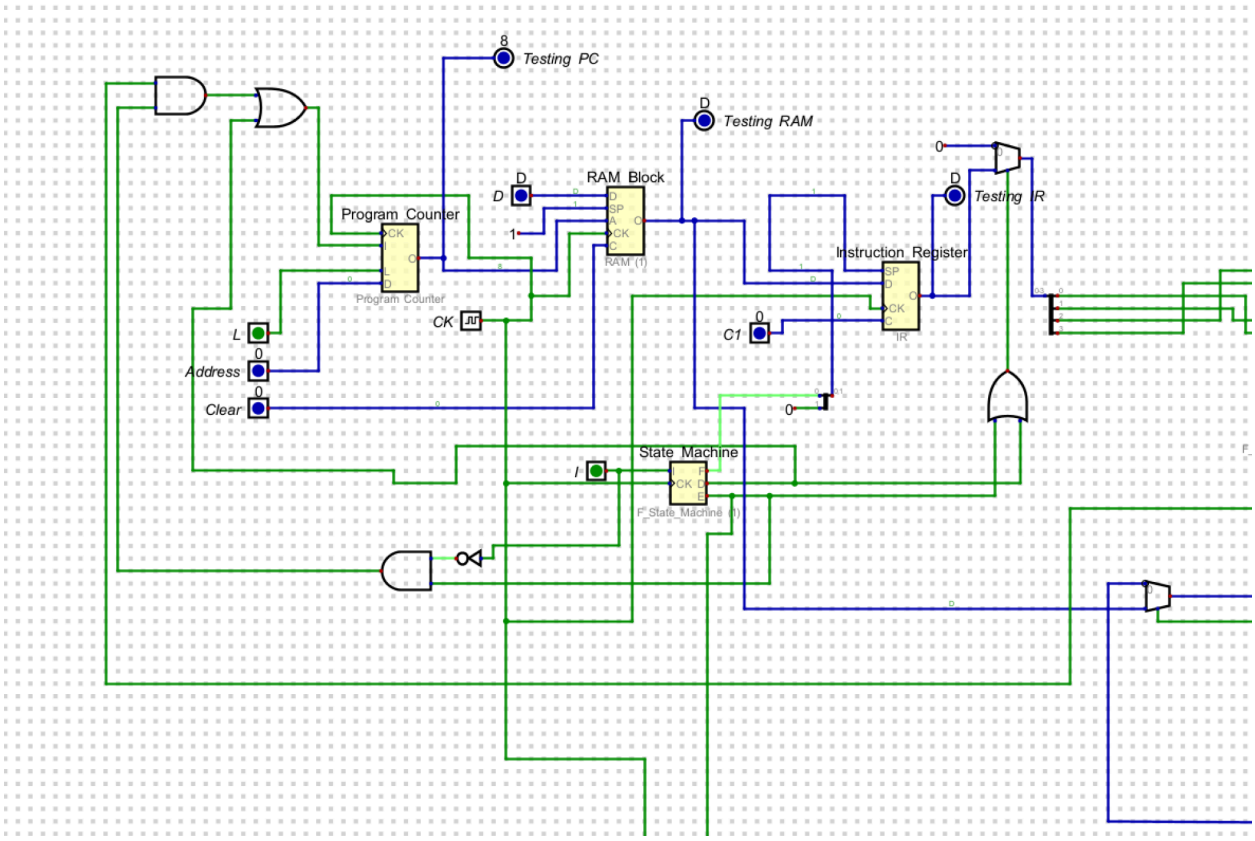
Starting PC - Address: 8

RAM Block Data: 13 (Opcode for XOR) - Now turn off I and write data 13 in the RAM Block

2 Clock Cycles: 1 for writing to RAM block, and 1 for loading value to IR - Go through 2 clock cycles to load it into the IR

I for Finite State Machine = 0 to stay in Fetch cycle - Keep off to load into IR so we can stay in the fetch cycle

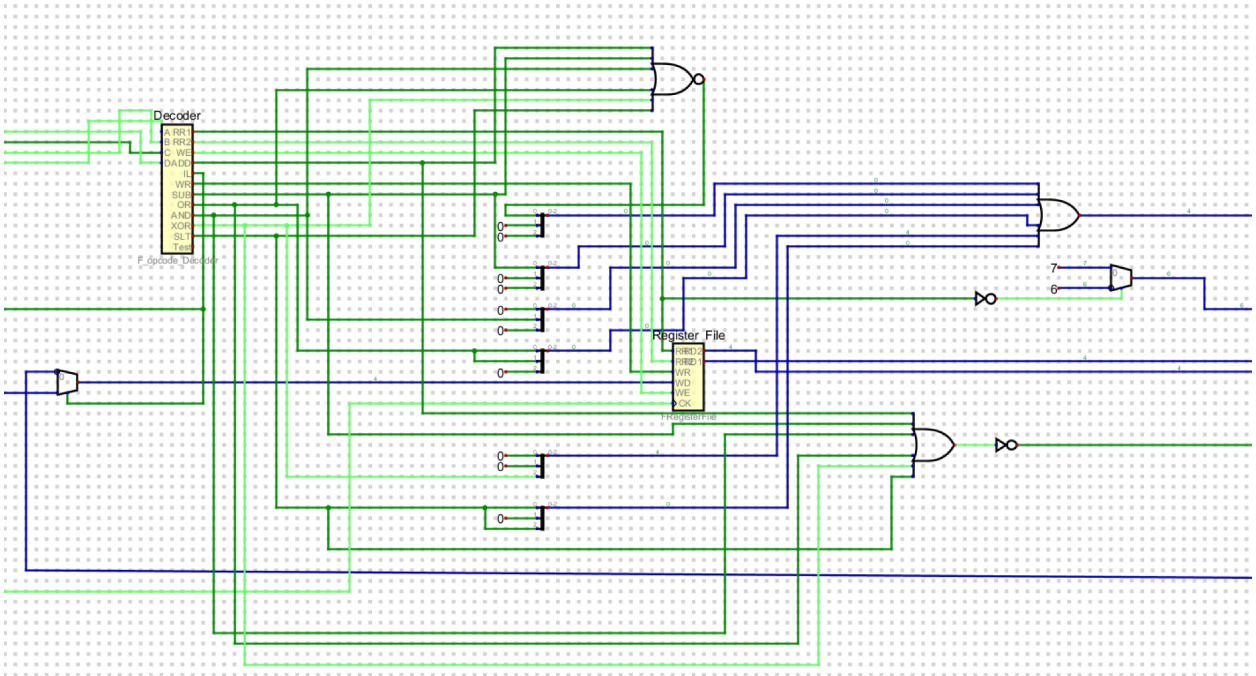
Then turn I = 1, go through 1 clock cycle to switch to decode cycle



Decode:

Now PC has incremented by 1, now we are not writing any immediate values

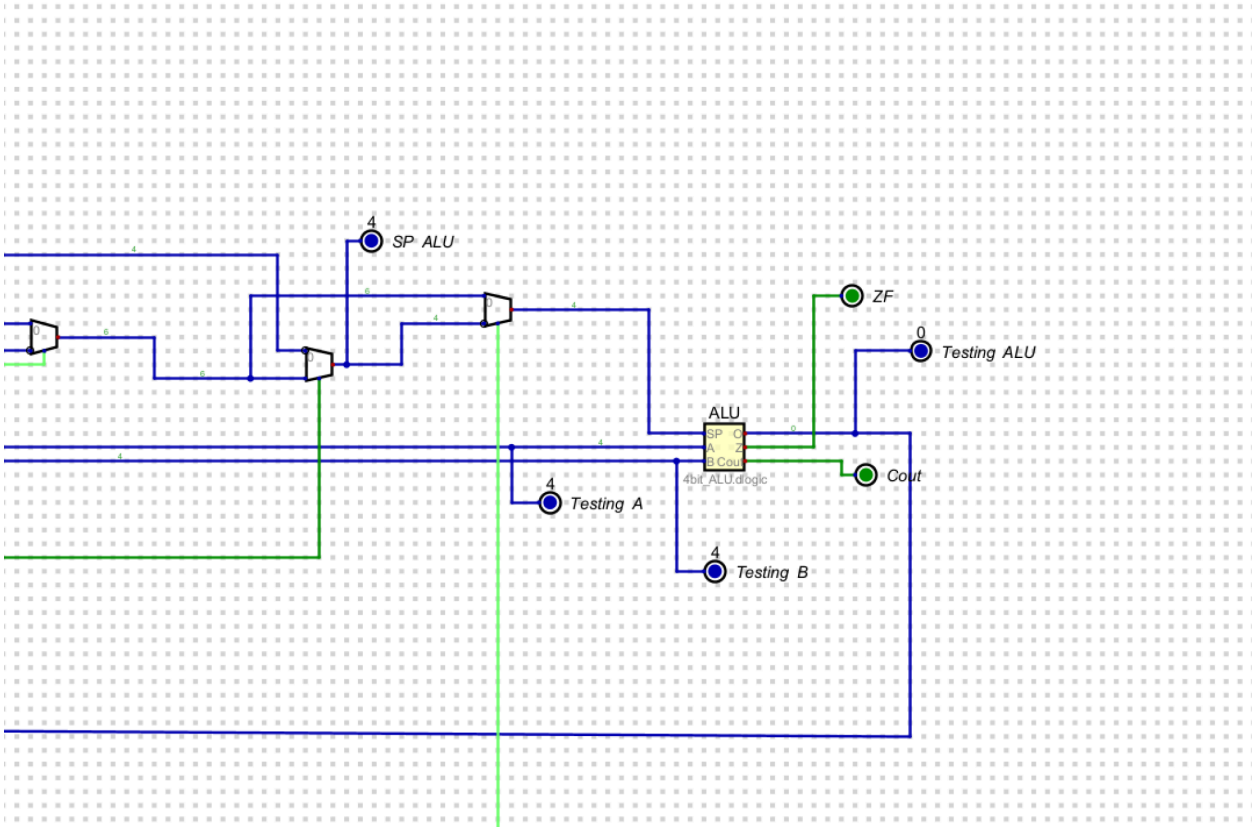
Keep I = 1, and do another clock cycle to enter execute cycle



Execute:

The ALU output is now 0 (4 XOR 4 = 0)

now go through the 1 clock cycle to go back to Fetch.



8.

SLT Opcode: 1110, Read Register 1: 0, Read Register 2: 1, Write Register: 0, Write Enable: Enabled, Immediate Load: Disabled, ALU Operation: SLT

Fetch:

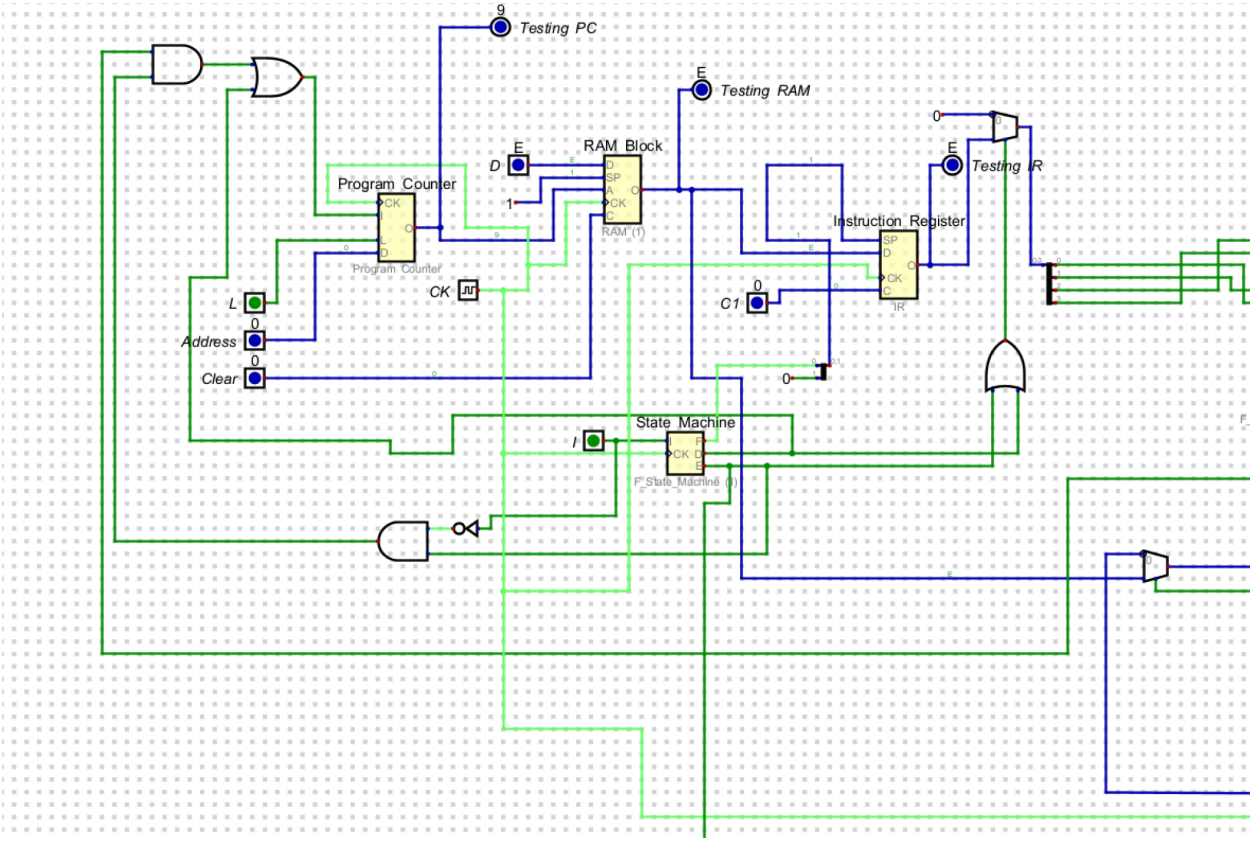
Starting PC - Address: 9

RAM Block Data: 14 (Opcode for SLT) - Now turn off I and write data 14 in the RAM Block

2 Clock Cycles: 1 for writing to RAM block, and 1 for loading value to IR - Go through 2 clock cycles to load it into the IR

I for Finite State Machine = 0 to stay in Fetch cycle - Keep off to load into IR so we can stay in the fetch cycle

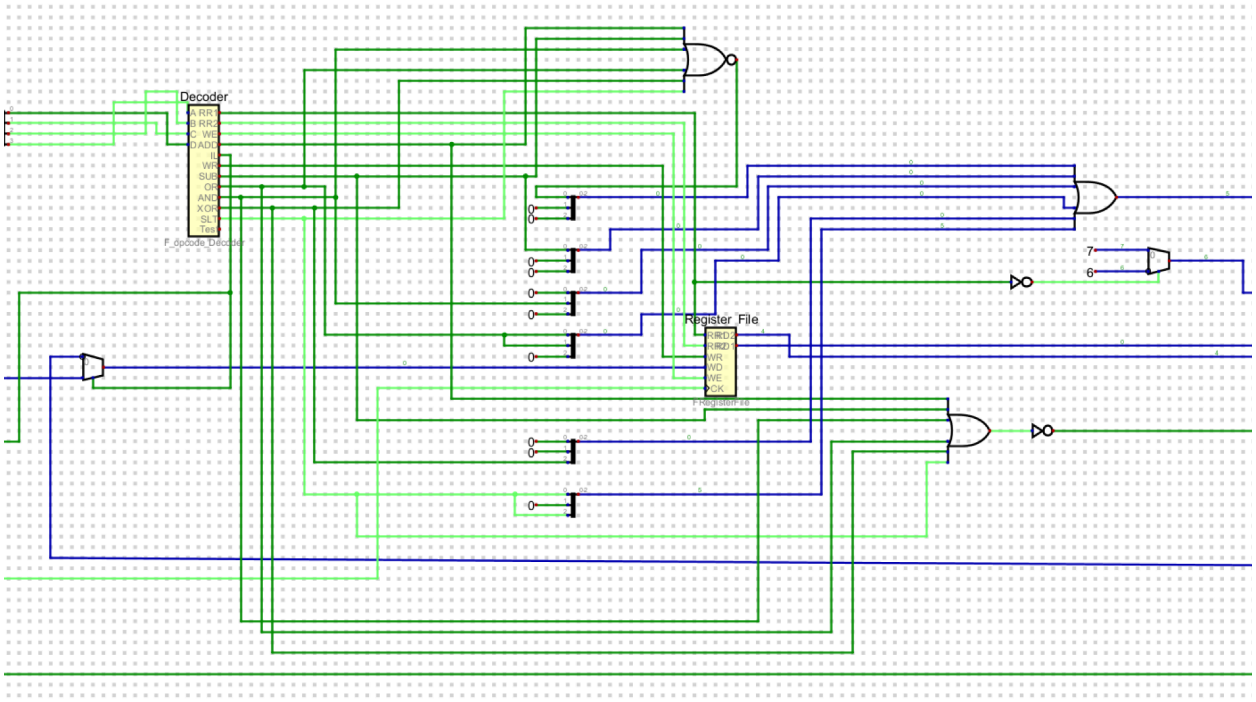
Then turn I = 1, go through 1 clock cycle to switch to decode cycle



Decode:

Now PC has incremented by 1, now we are not writing any immediate values

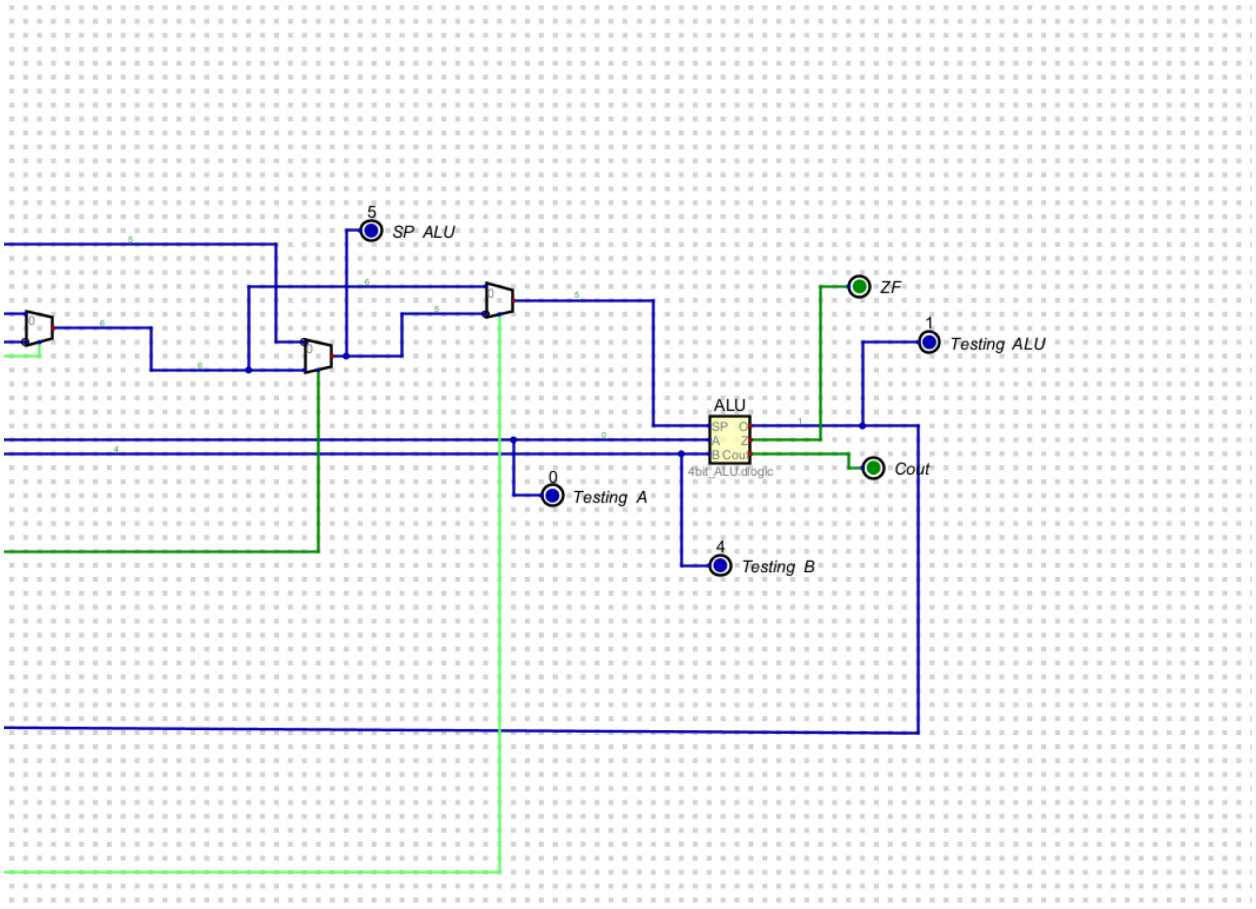
Keep I = 1, and do another clock cycle to enter execute cycle



Execute:

The ALU output is now 1 (0 SLT 4 = 1)

now go through the 1 clock cycle to go back to Fetch.



9.

No Operation (NOP) Opcode: 0000, Write Enable: Disabled, Immediate Load: Disabled, ALU Operation: No Operation

Fetch:

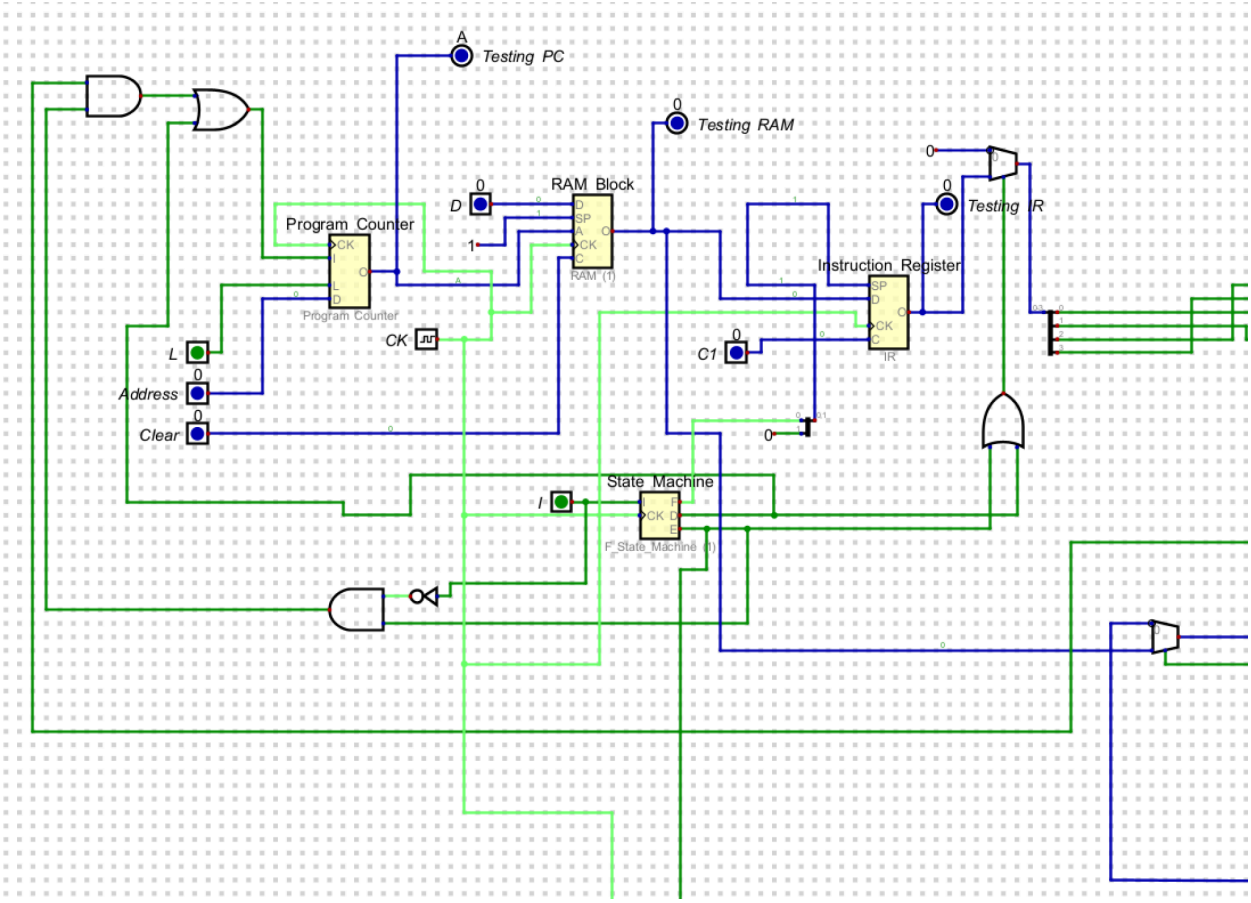
Starting PC - Address: 10

RAM Block Data: 0 (Opcode for NOP) - Now turn off I and write data 0 in the RAM Block

2 Clock Cycles: 1 for writing to RAM block, and 1 for loading value to IR - Go through 2 clock cycles to load it into the IR

I for Finite State Machine = 0 to stay in Fetch cycle - Keep off to load into IR so we can stay in the fetch cycle

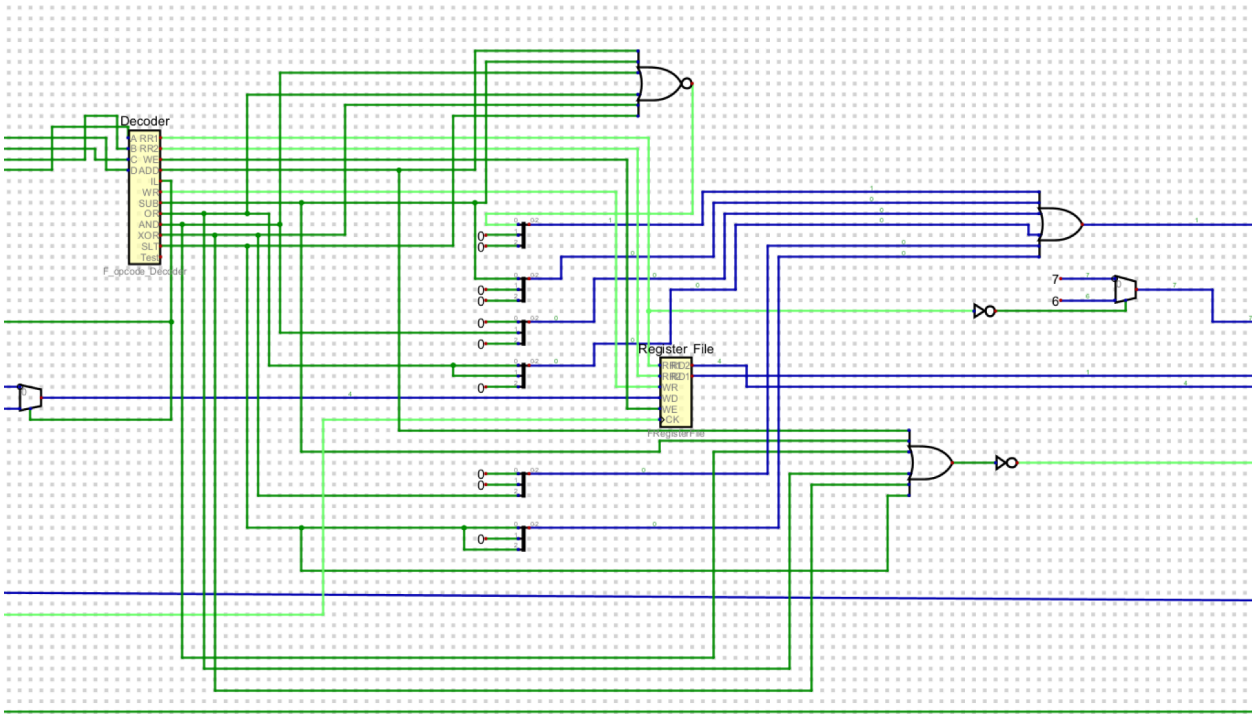
Then turn I = 1, go through 1 clock cycle to switch to decode cycle



Decode:

Now PC has incremented by 1, now we are not writing any immediate values

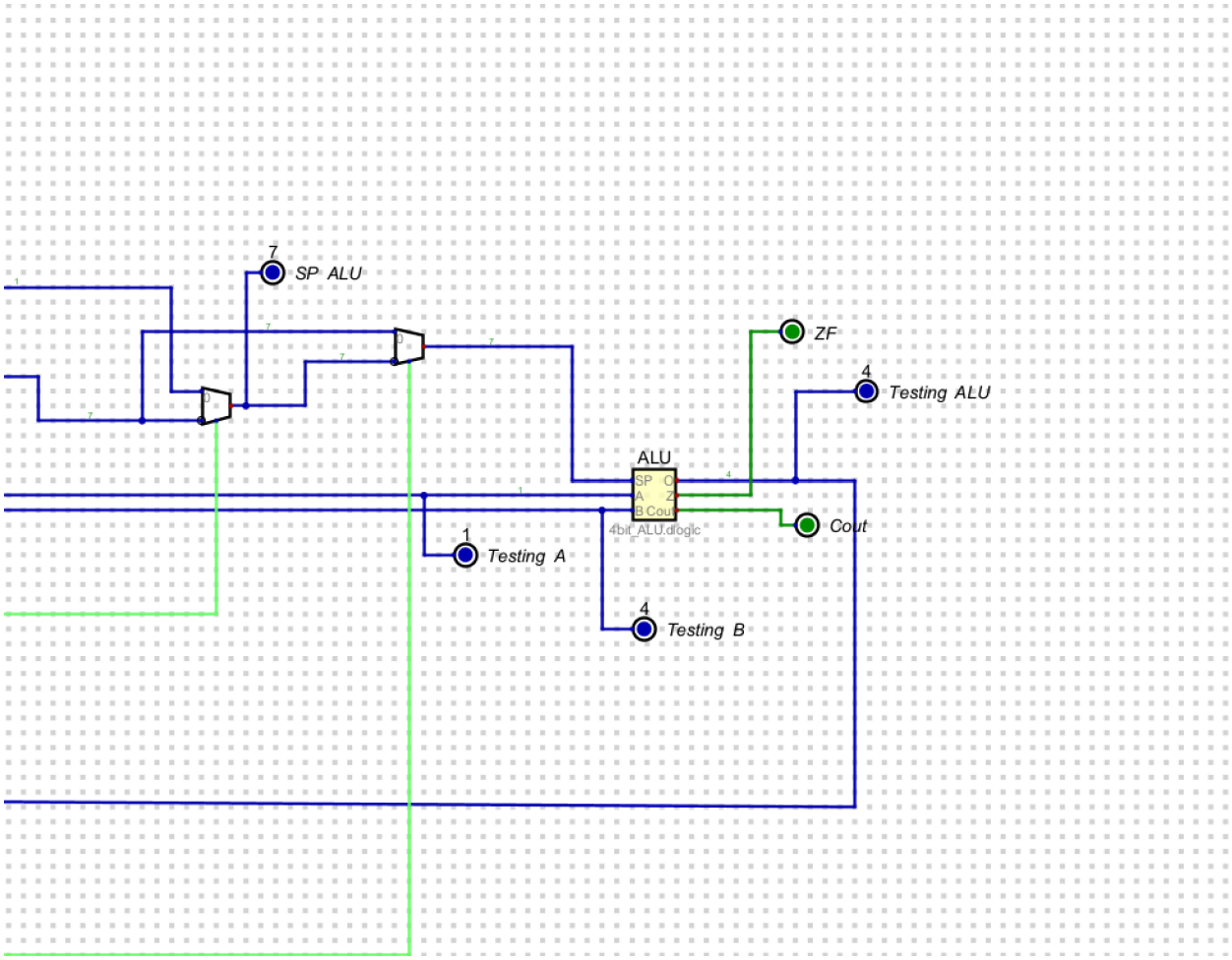
Keep I = 1, and do another clock cycle to enter execute cycle



Execute:

The ALU output is 4 (1 NOP 4 = 4) No Change

now go through the 1 clock cycle to go back to Fetch.



10.

MOV A,B (B to A) Opcode: 0111, Read Register 1: 1, Write Register: 0, Write Enable: Enabled, Immediate Load: Disabled, ALU Operation: No Operation

Fetch:

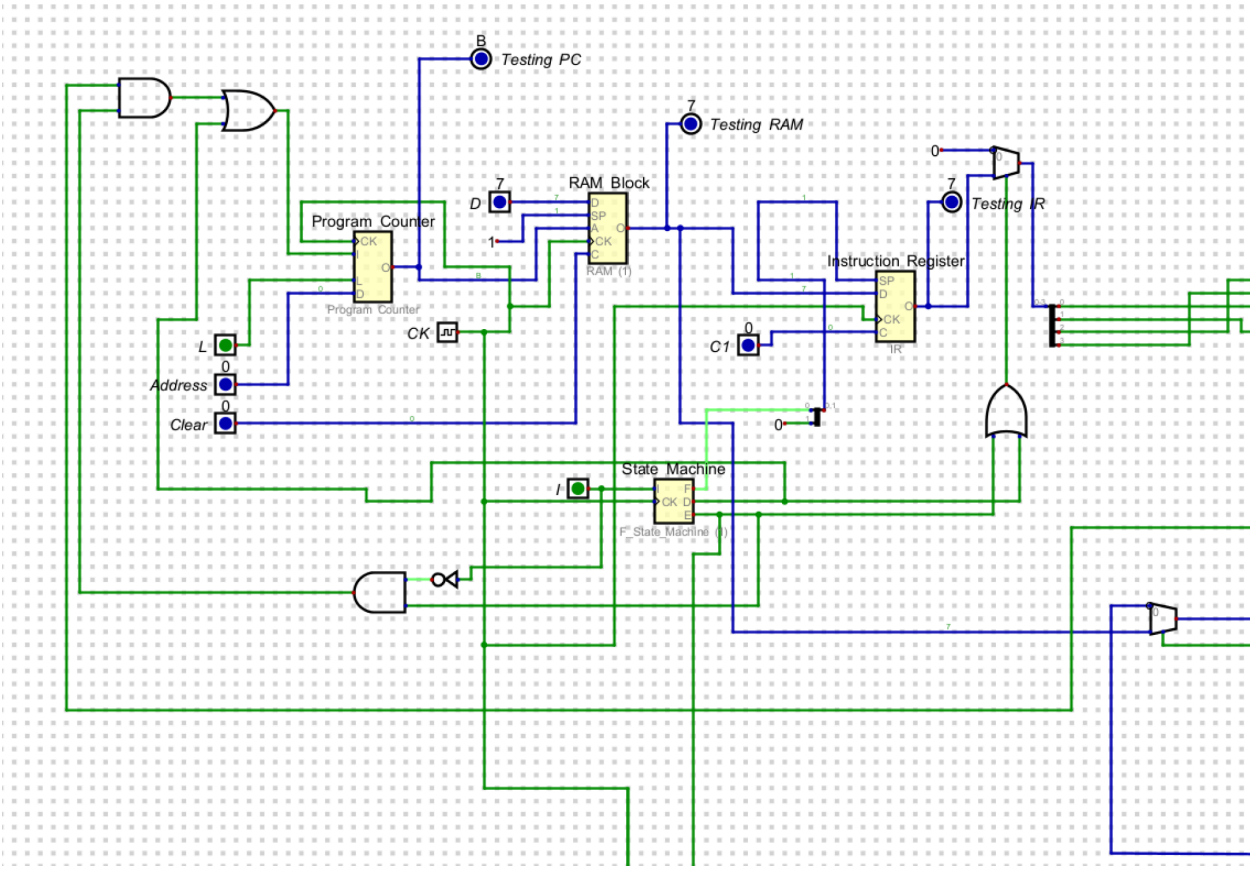
Starting PC - Address: 11

RAM Block Data: 7 (Opcode for MOV A,B) - Now turn off I and write data 7 in the RAM Block

2 Clock Cycles: 1 for writing to RAM block, and 1 for loading value to IR - Go through 2 clock cycles to load it into the IR

I for Finite State Machine = 0 to stay in Fetch cycle - Keep off to load into IR so we can stay in the fetch cycle

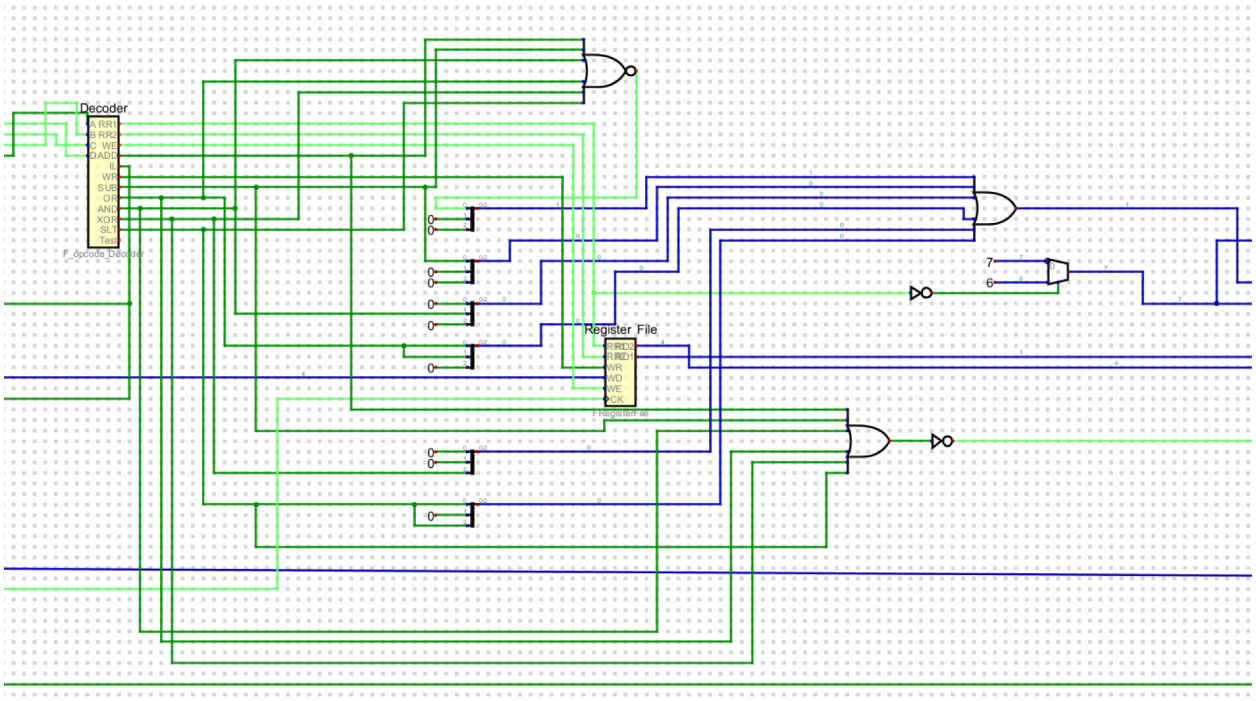
Then turn I = 1, go through 1 clock cycle to switch to decode cycle



Decode:

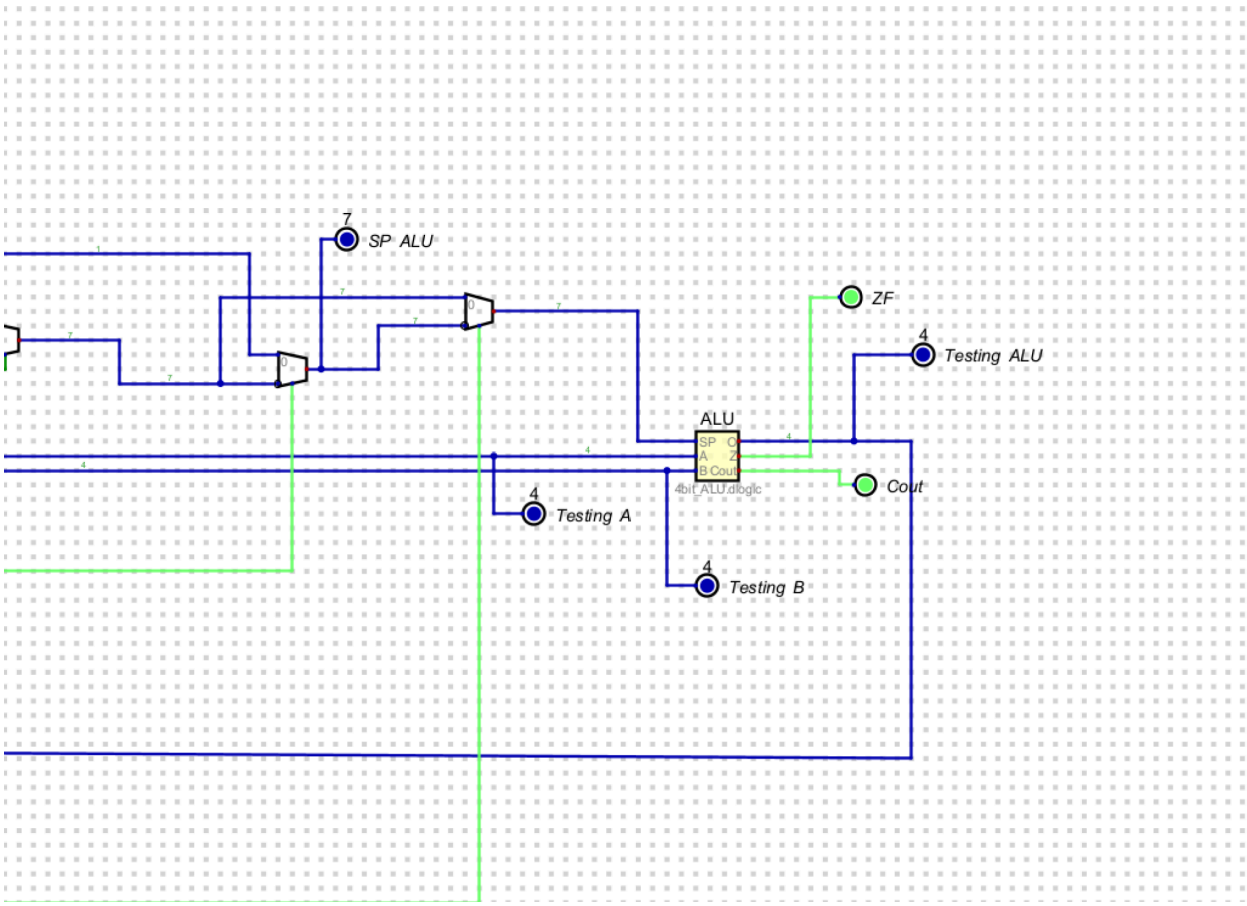
Now PC has incremented by 1, now we are not writing any immediate values

Keep I = 1, and do another clock cycle to enter execute cycle



Execute:

Register A will now have the same value as Register B which is 4 now go through the 1 clock cycle to go back to Fetch.



11.

Since Register A and Register B have the same values, immediate load a different value to B so you can test the last operation which is MOV B,A

MOV B,A (A to B) Opcode: 1000, Read Register 1: 0, Write Register: 1, Write Enable: Enabled, Immediate Load: Disabled, ALU Operation: No Operation

Fetch:

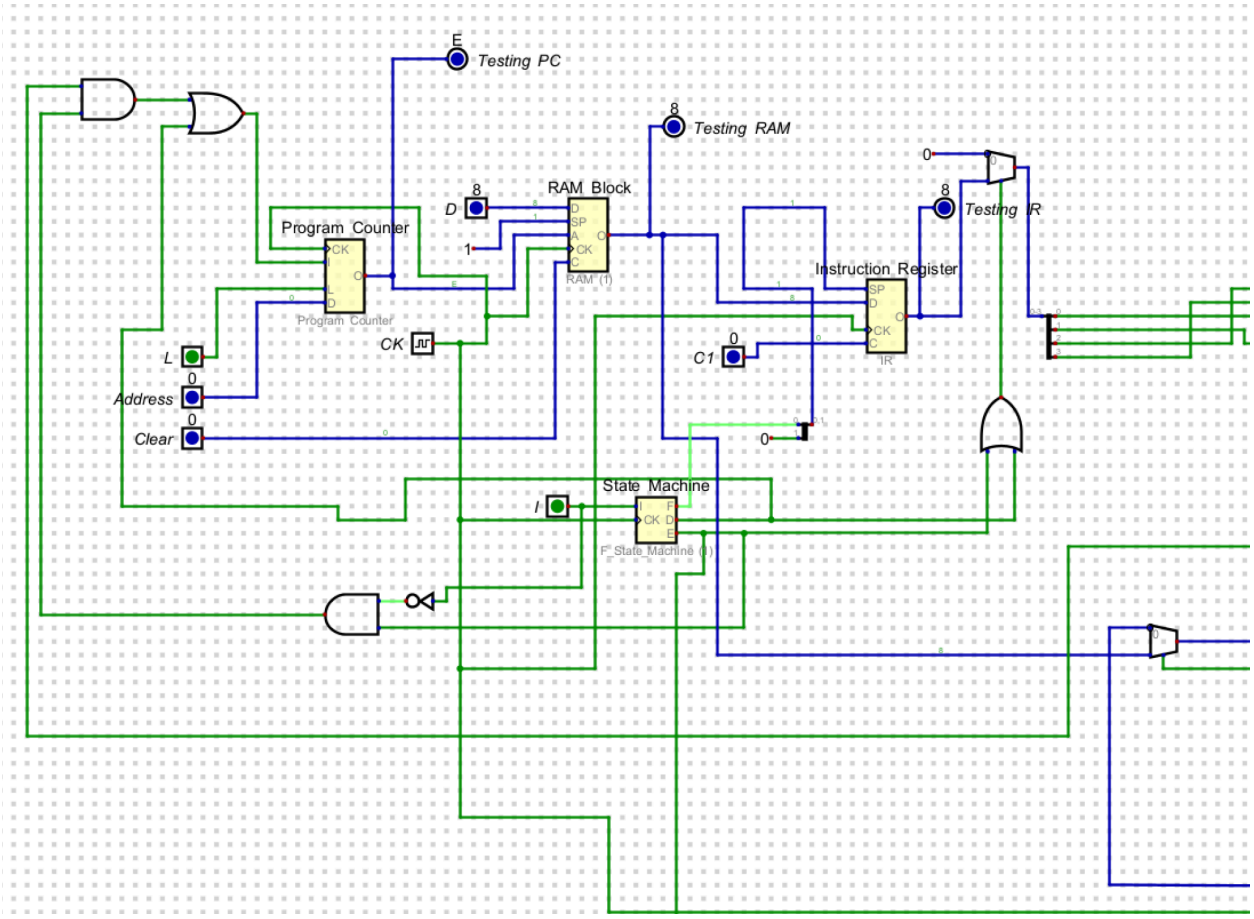
Starting PC - Address: 14

RAM Block Data: 8 (Opcode for MOV B,A) - Now turn off I and write data 8 in the RAM Block

2 Clock Cycles: 1 for writing to RAM block, and 1 for loading value to IR - Go through 2 clock cycles to load it into the IR

I for Finite State Machine = 0 to stay in Fetch cycle - Keep off to load into IR so we can stay in the fetch cycle

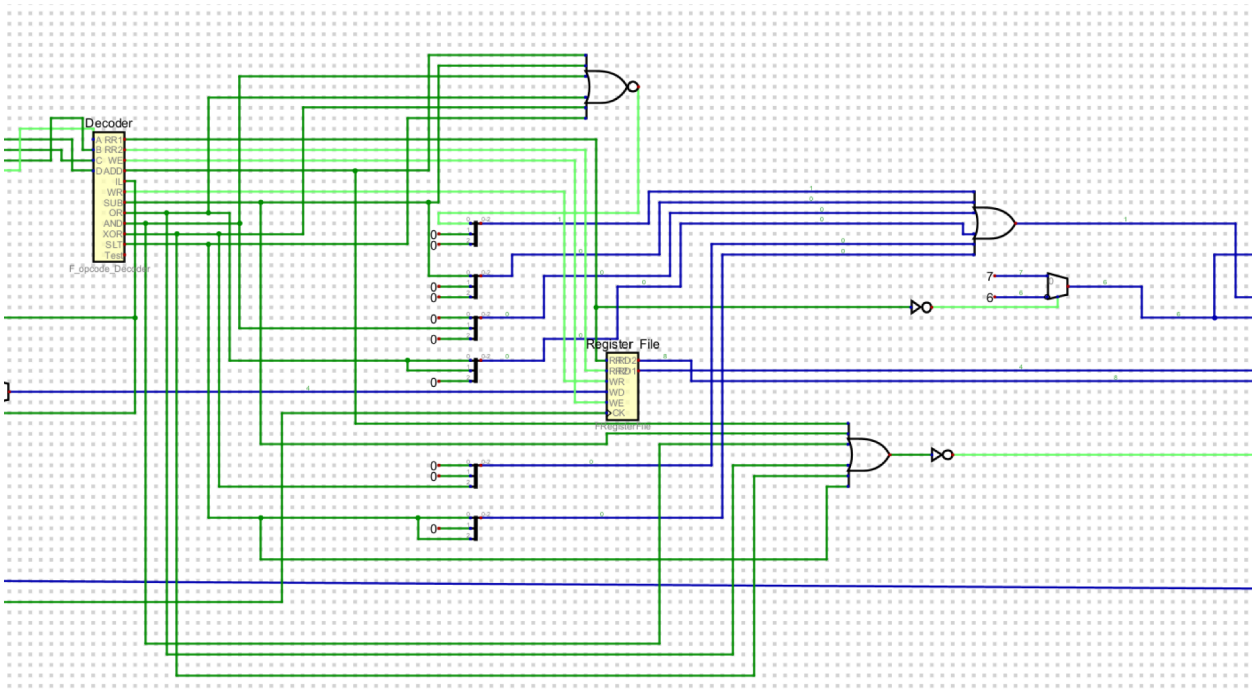
Then turn I = 1, go through 1 clock cycle to switch to decode cycle



Decode:

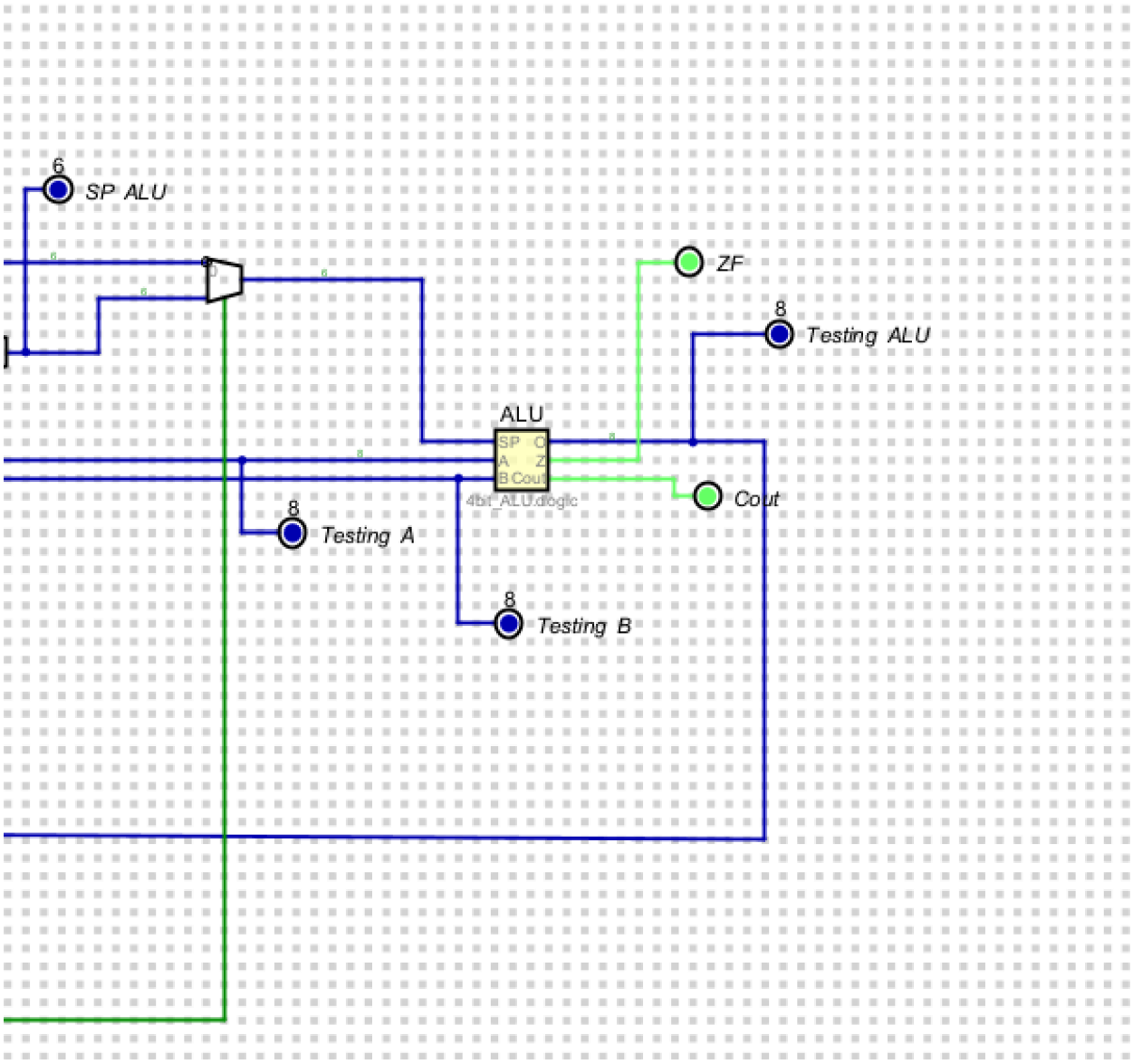
Now PC has incremented by 1, now we are not writing any immediate values

Keep I = 1, and do another clock cycle to enter execute cycle



Execute:

Register B will now have the same value as Register A now go through the 1 clock cycle to go back to Fetch.


6
SP ALU
ZF
8
Testing ALU
ALU
SP O
A Z
B Cout
4bit_ALU.diogle
Cout
8
Testing A
8
Testing B