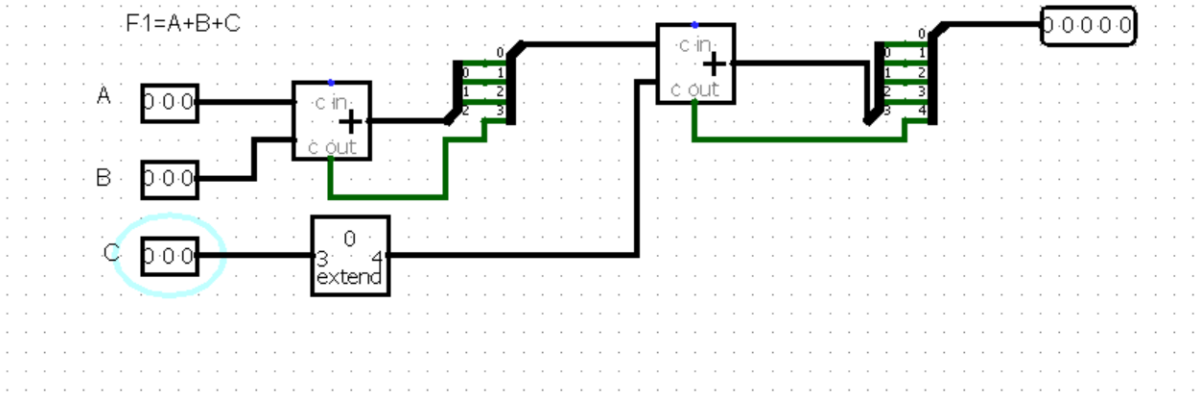
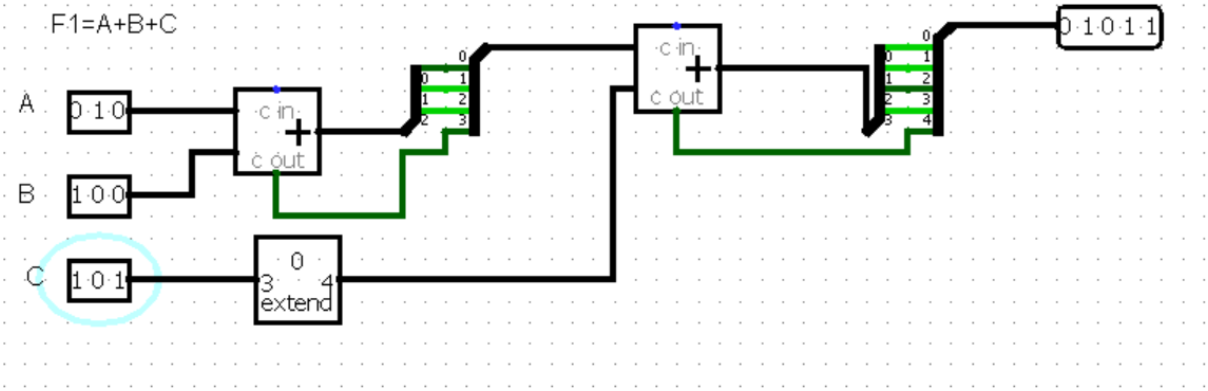
Part 1: Implementation of the components separately

F1= A+B+C



Testing of function F1 is given below:

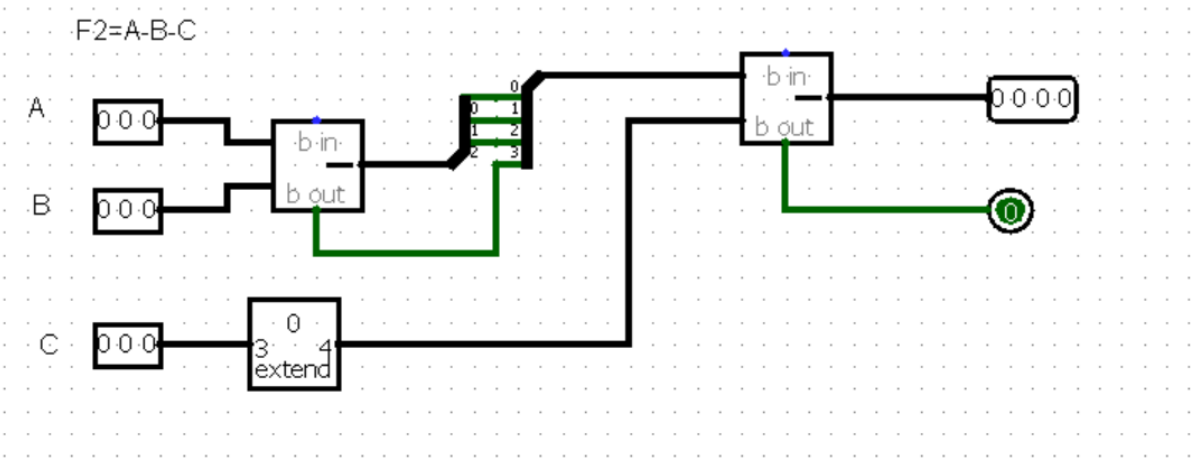


This testing is correct as the answer is same when it was executed manually.

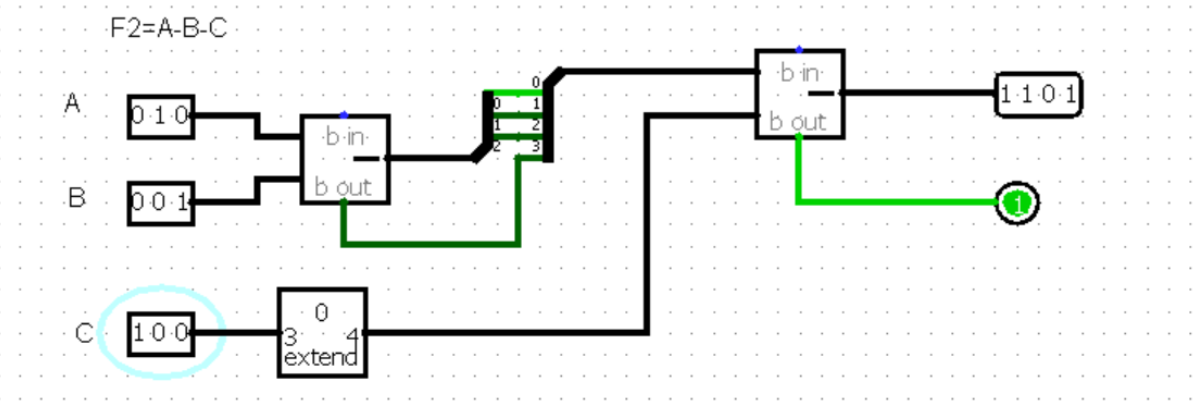
```
  010
 +100
  110

  110
 +101
 1011
```

F2= A-B-C



Testing of function F2 is given below:



This testing is correct as it gave same result when it was executed manually.

010

-001 = 110 (Complemented the small number)

+1

111

+010

1001 1001

-100 = 1011 (Complemented the small number)

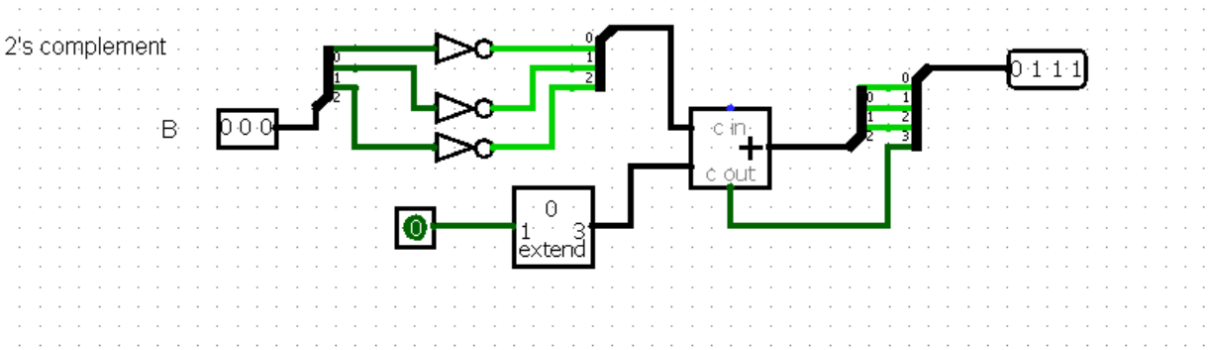
+1

1100

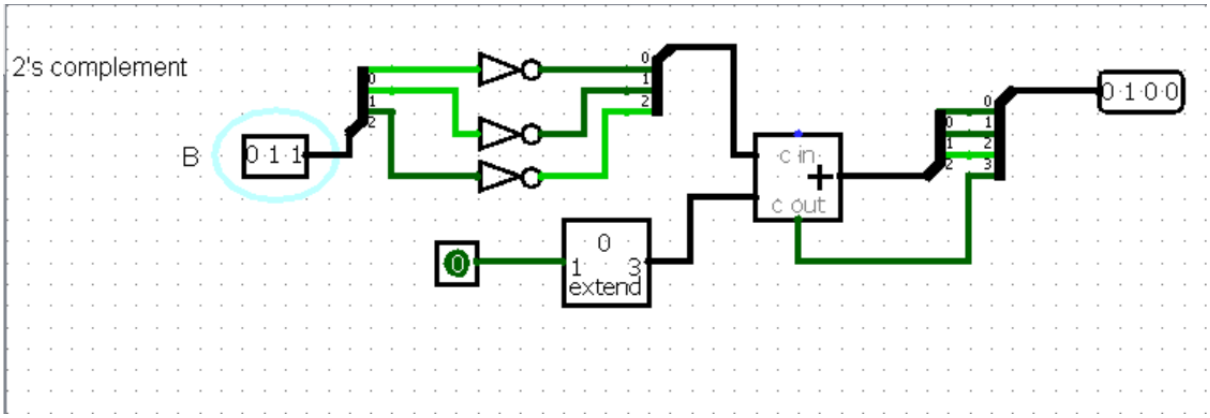
+001

1101

F3= 2's complement of B



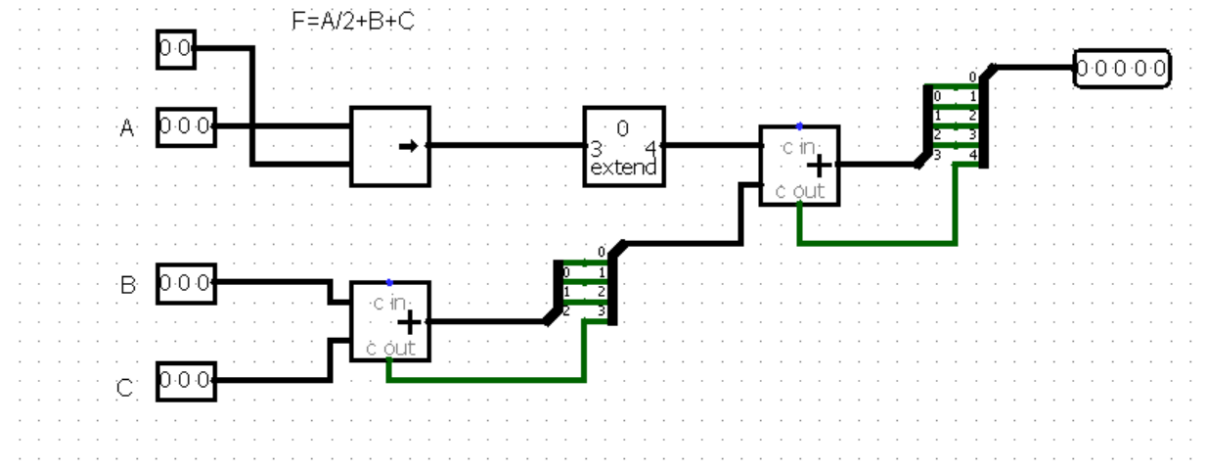
Testing of function F3 is given below:



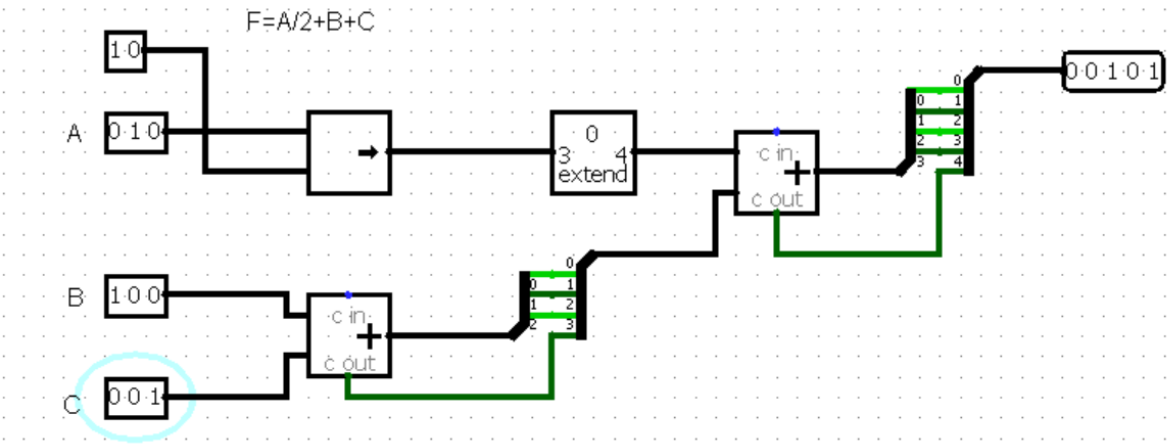
This result is accurate as it gave the same result when executed manually.

011 gives 100 when complemented.

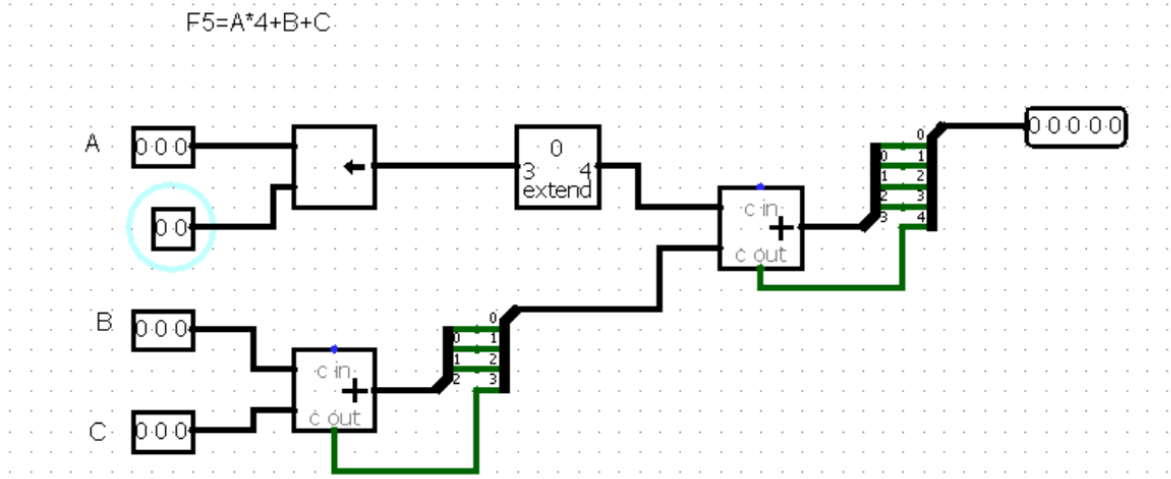
F4= A/2+B+C



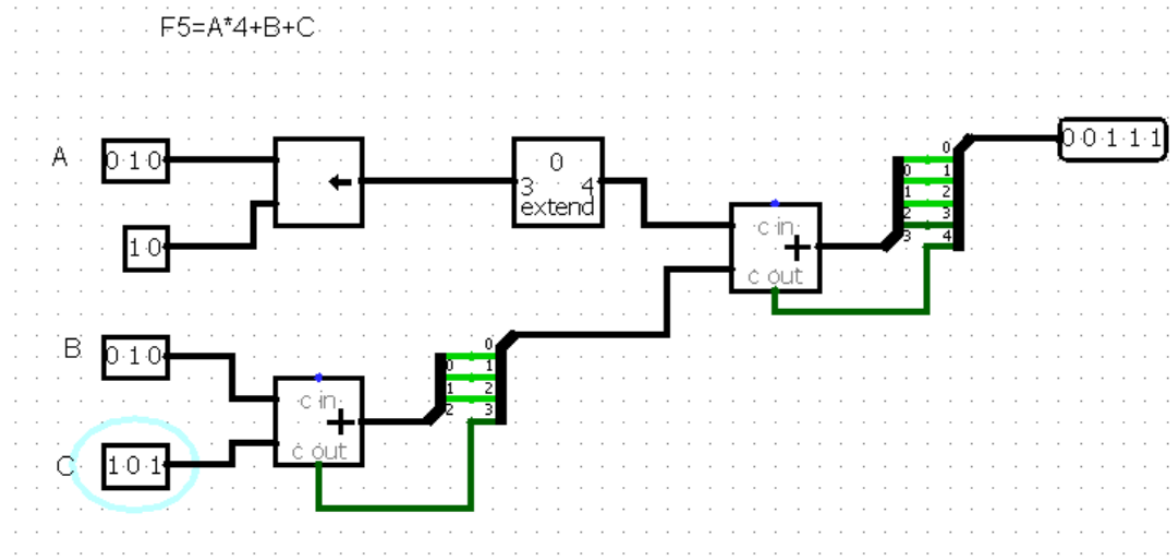
Testing of function F4 is given below:



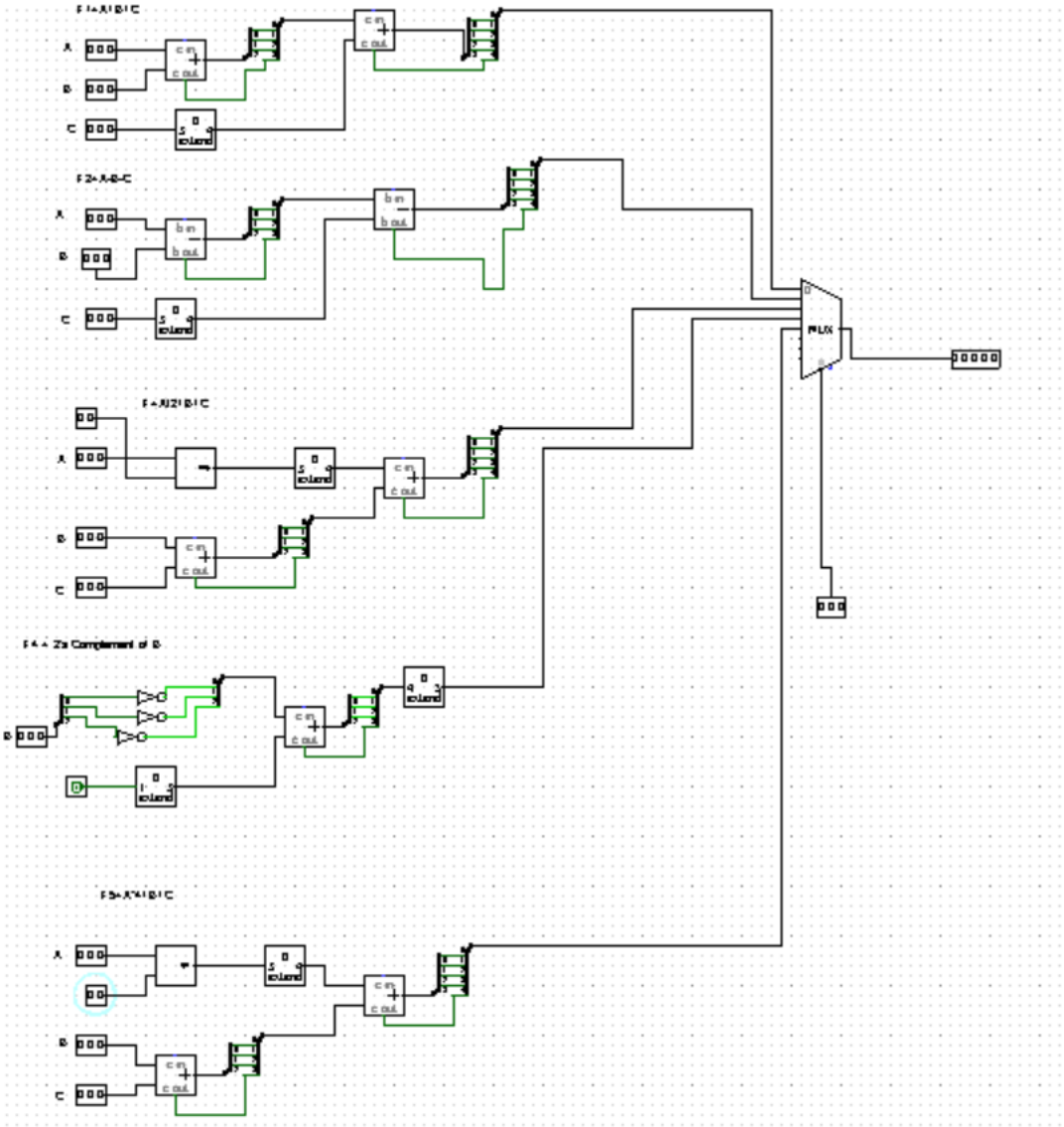
F5= A*4+B+C



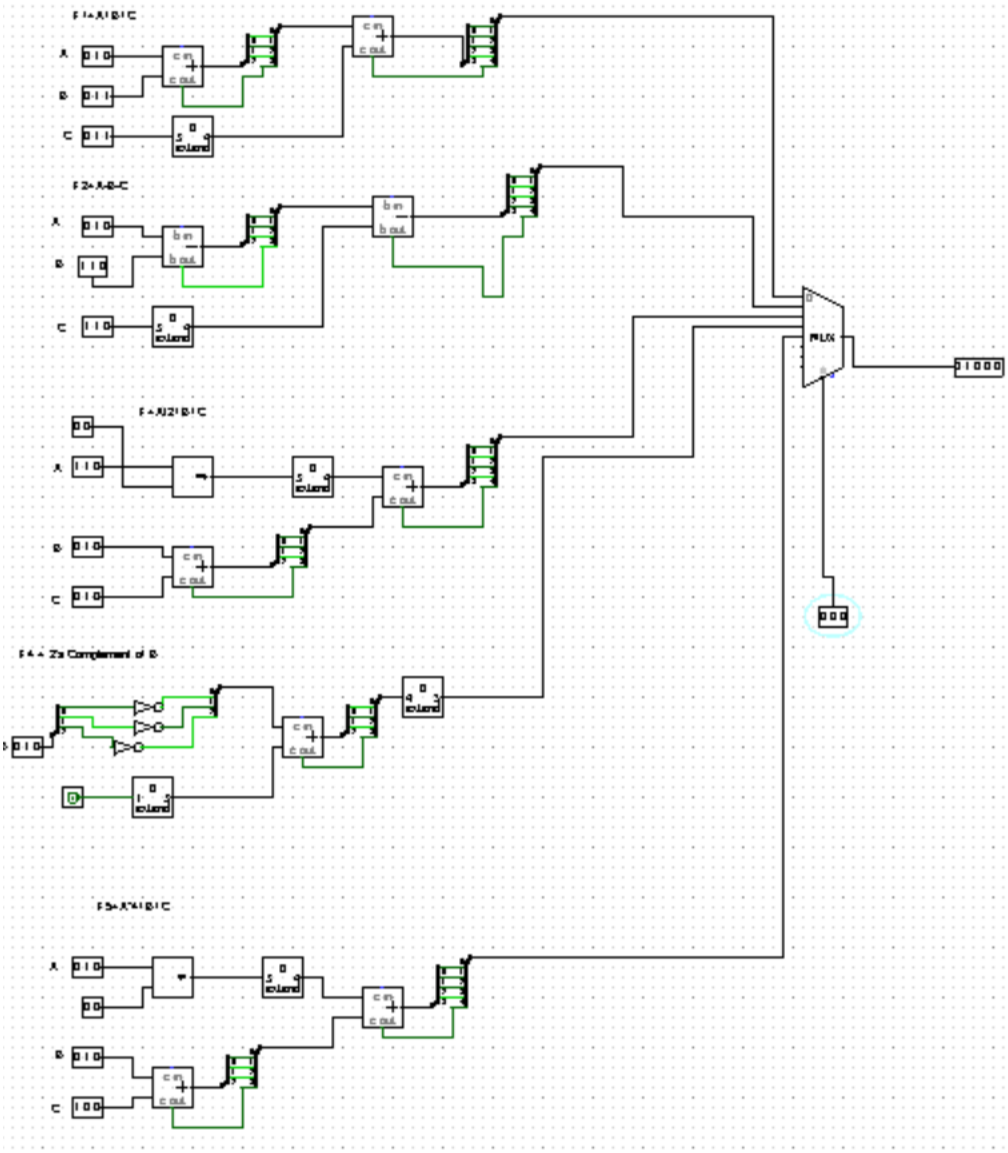
Testing of function F5 is given below:



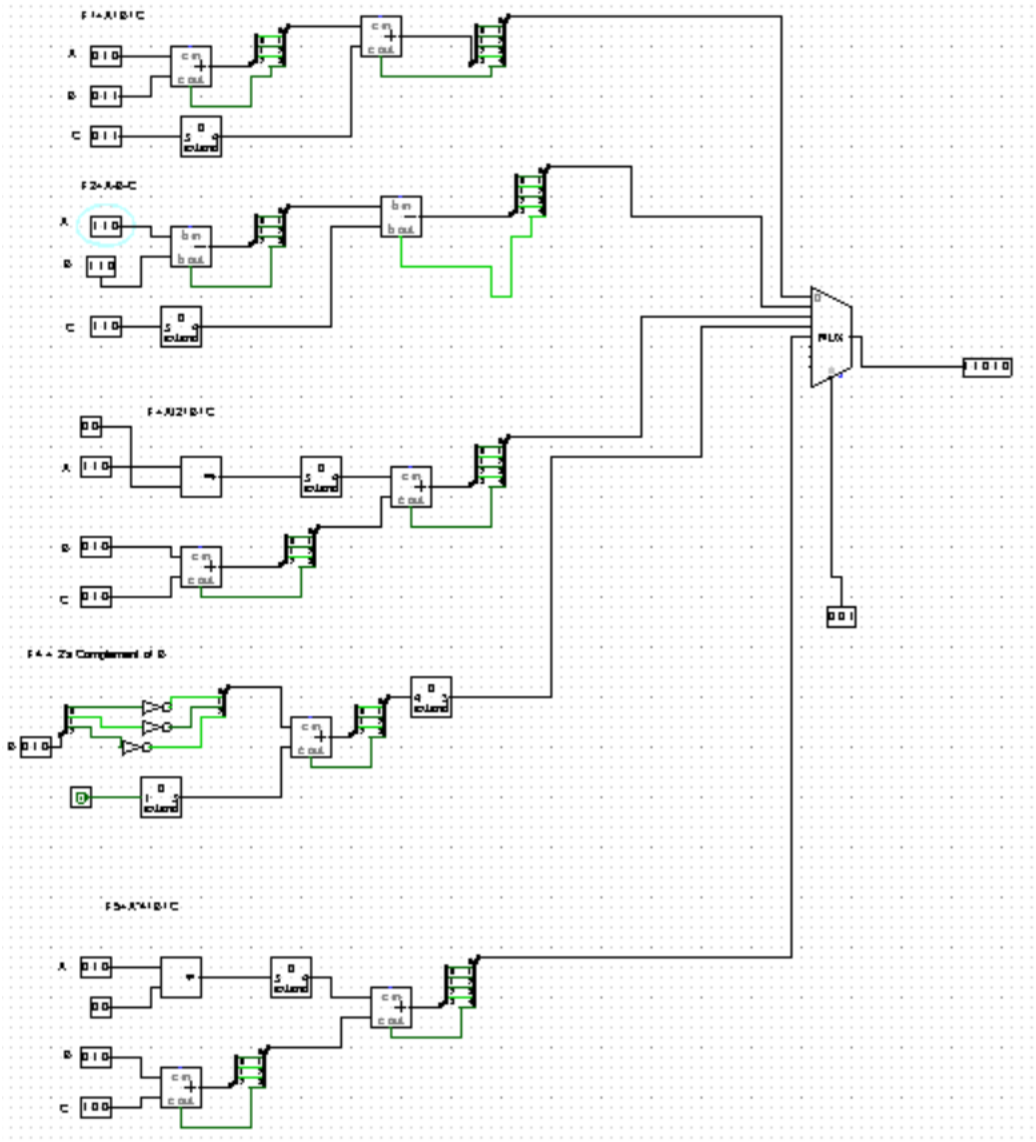
Part 2: Integration of all five components to build an ALU



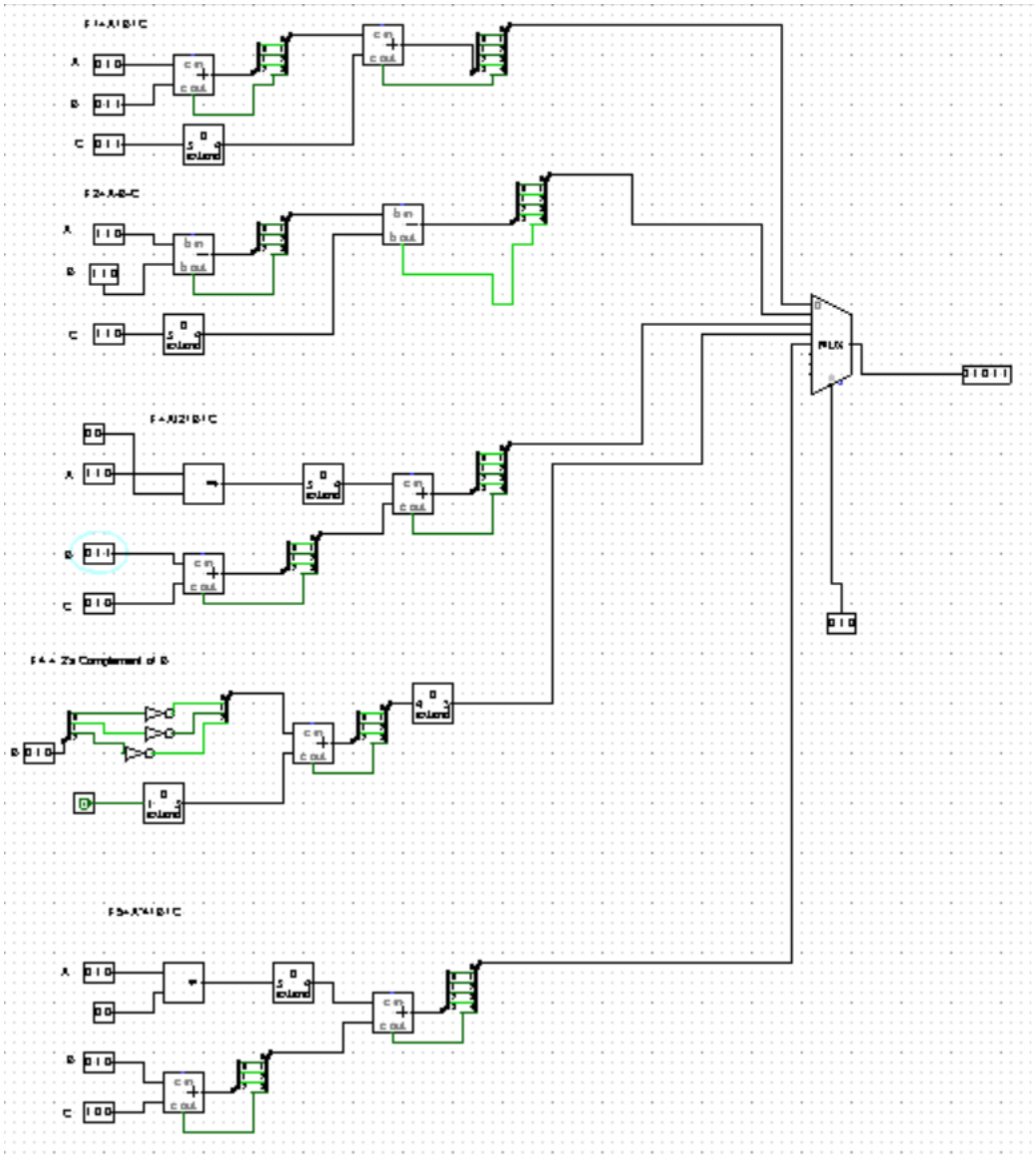
Testing 1:



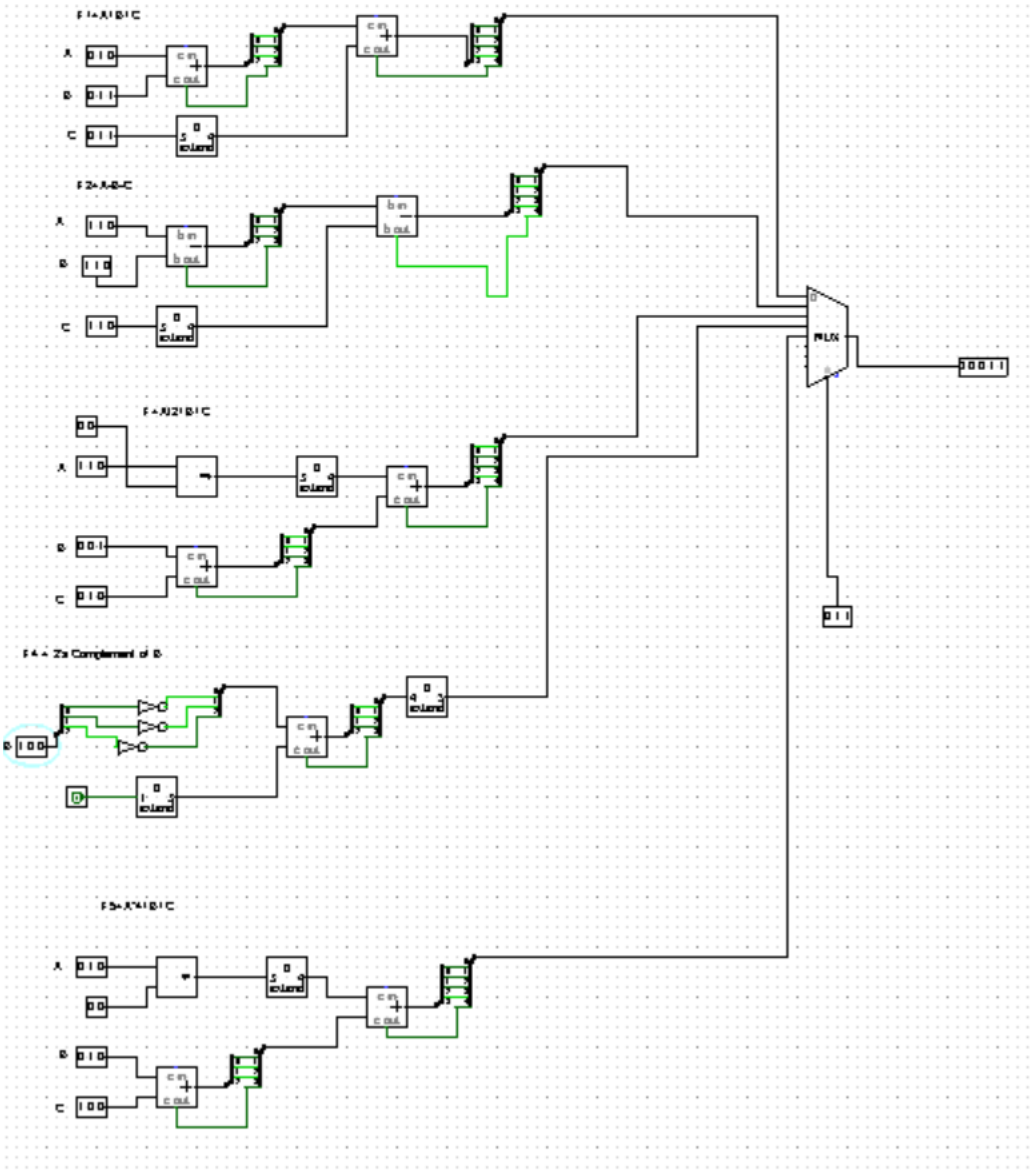
Testing 2:


F1=A+B+C
A
B
C
F2=A-B-C
F3=A*2+B+C
F4=2's Complement of B
F5=A*4+B+C
MUX
1010
001

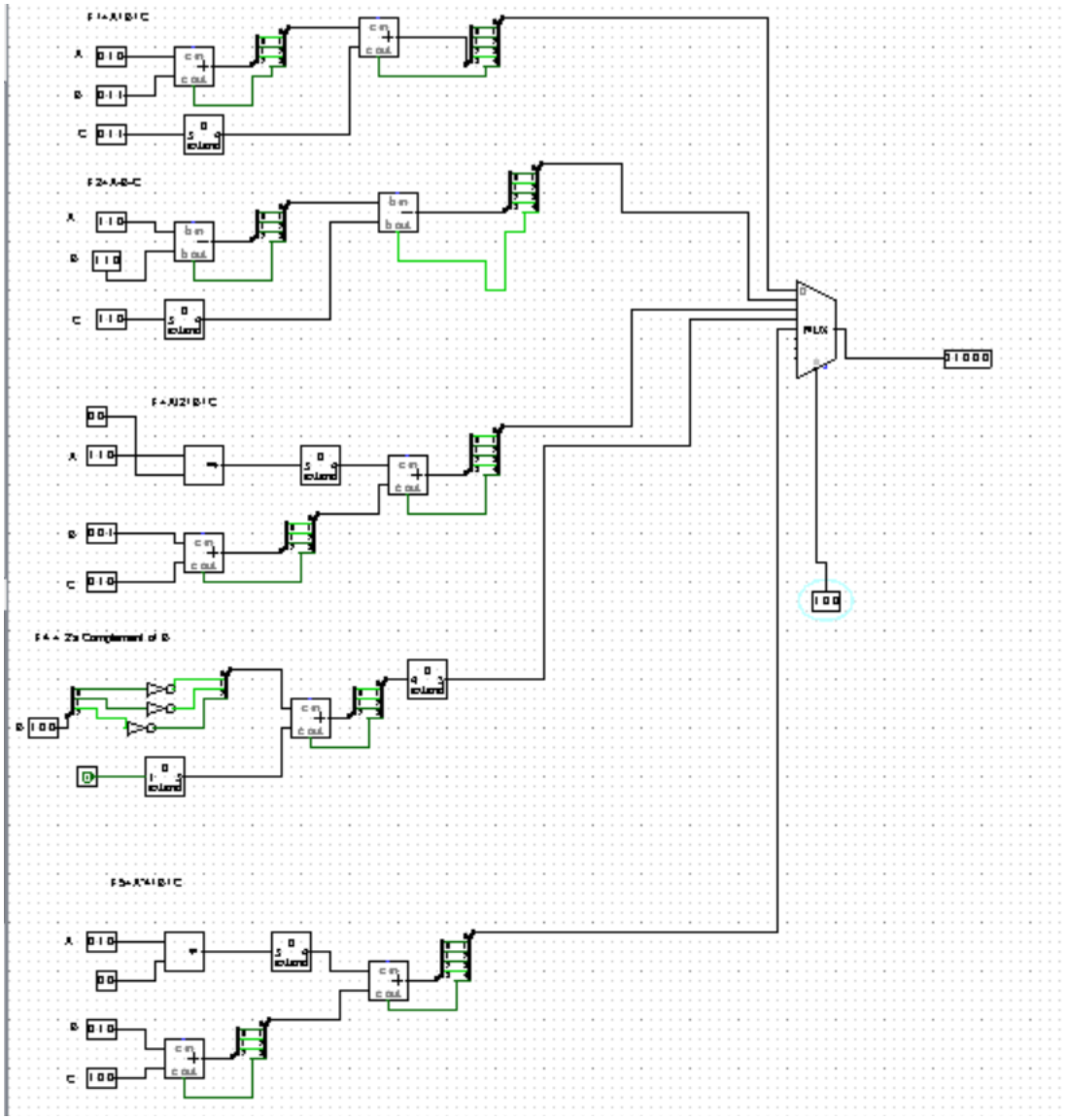
Testing 3:



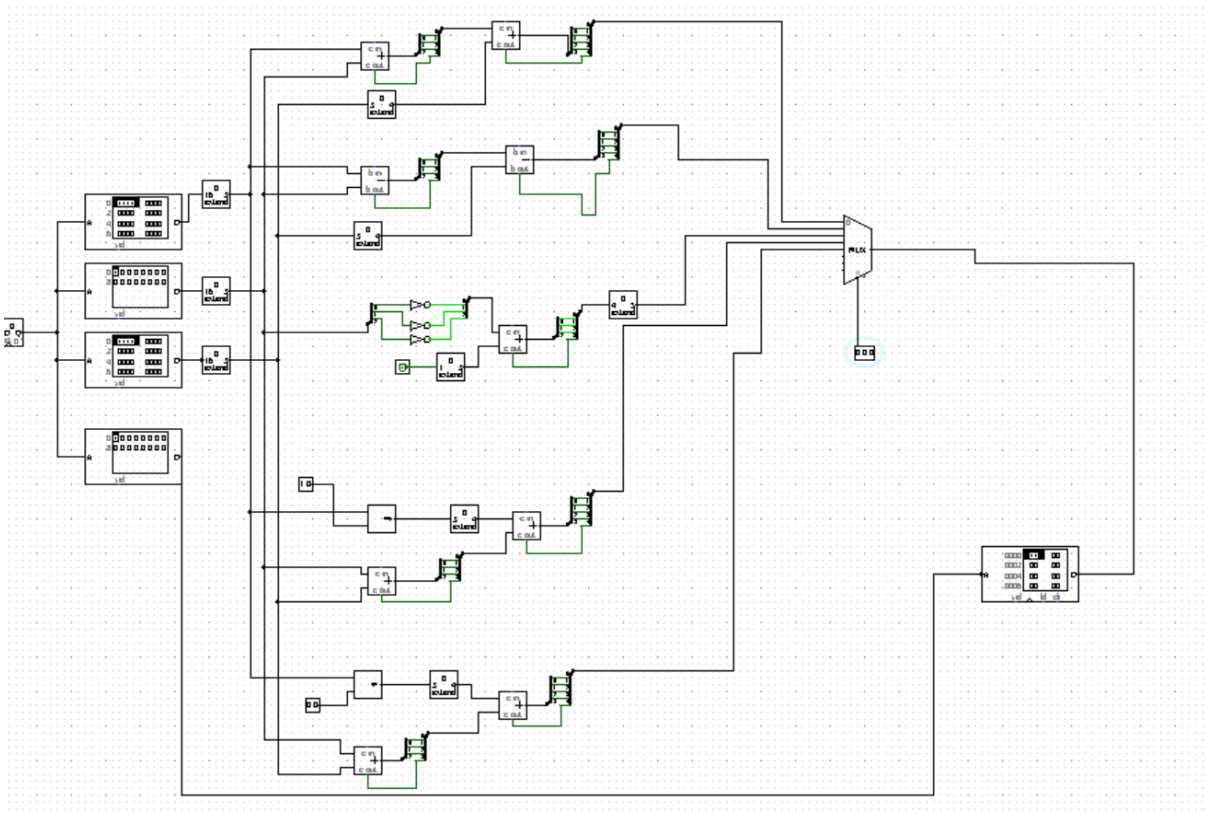
Testing 4:


F1=A+B+C
F2=A-B-C
F3=A+2+B+C
F4= 2's Complement of B
F5=A*4+B+C
MUX
00011
011

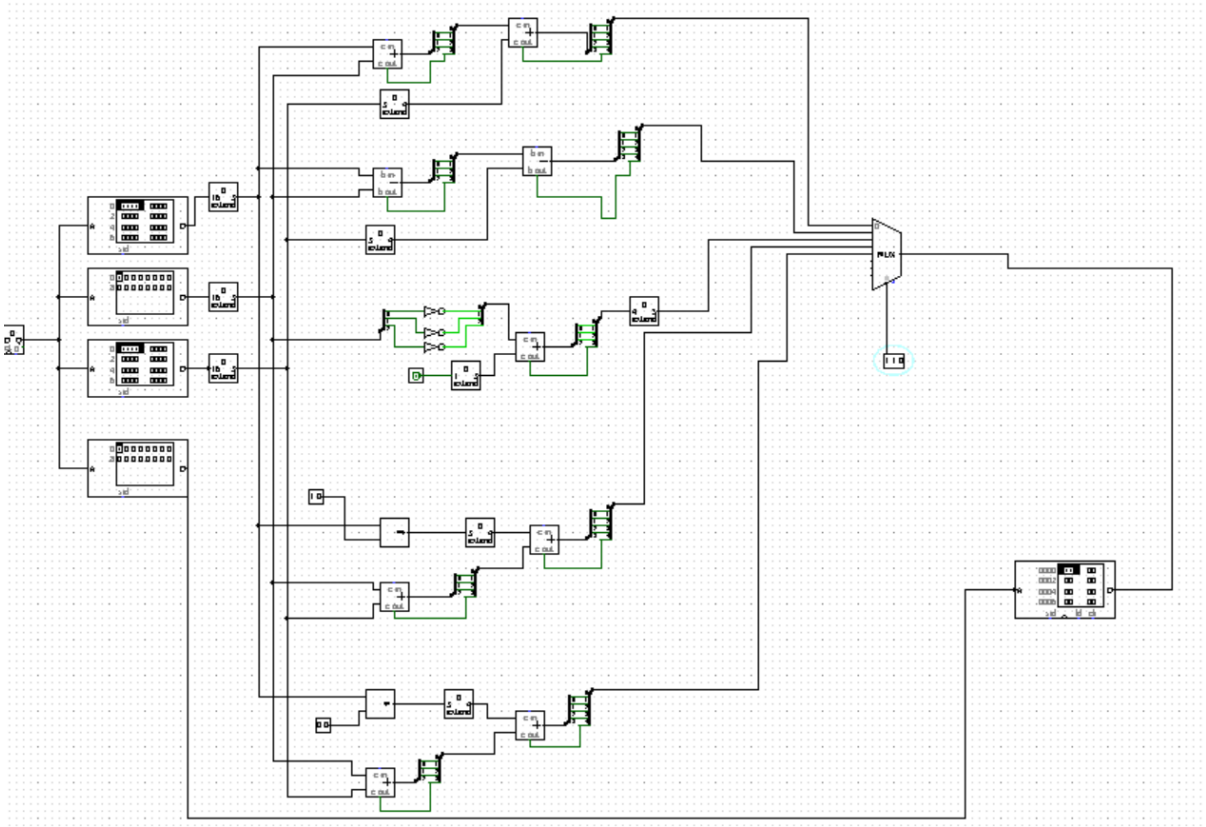
Testing 5:



Part 3: A toy computer built using ALU, inbuilt ROMS and RAM:

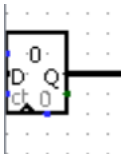


Testing of part 3 is given below:

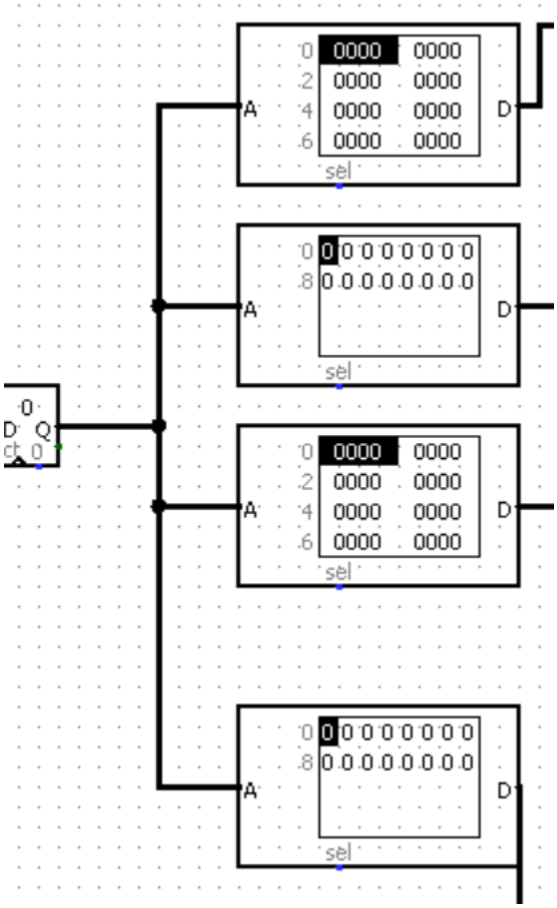


Creation of General Purpose Computer using ALU :

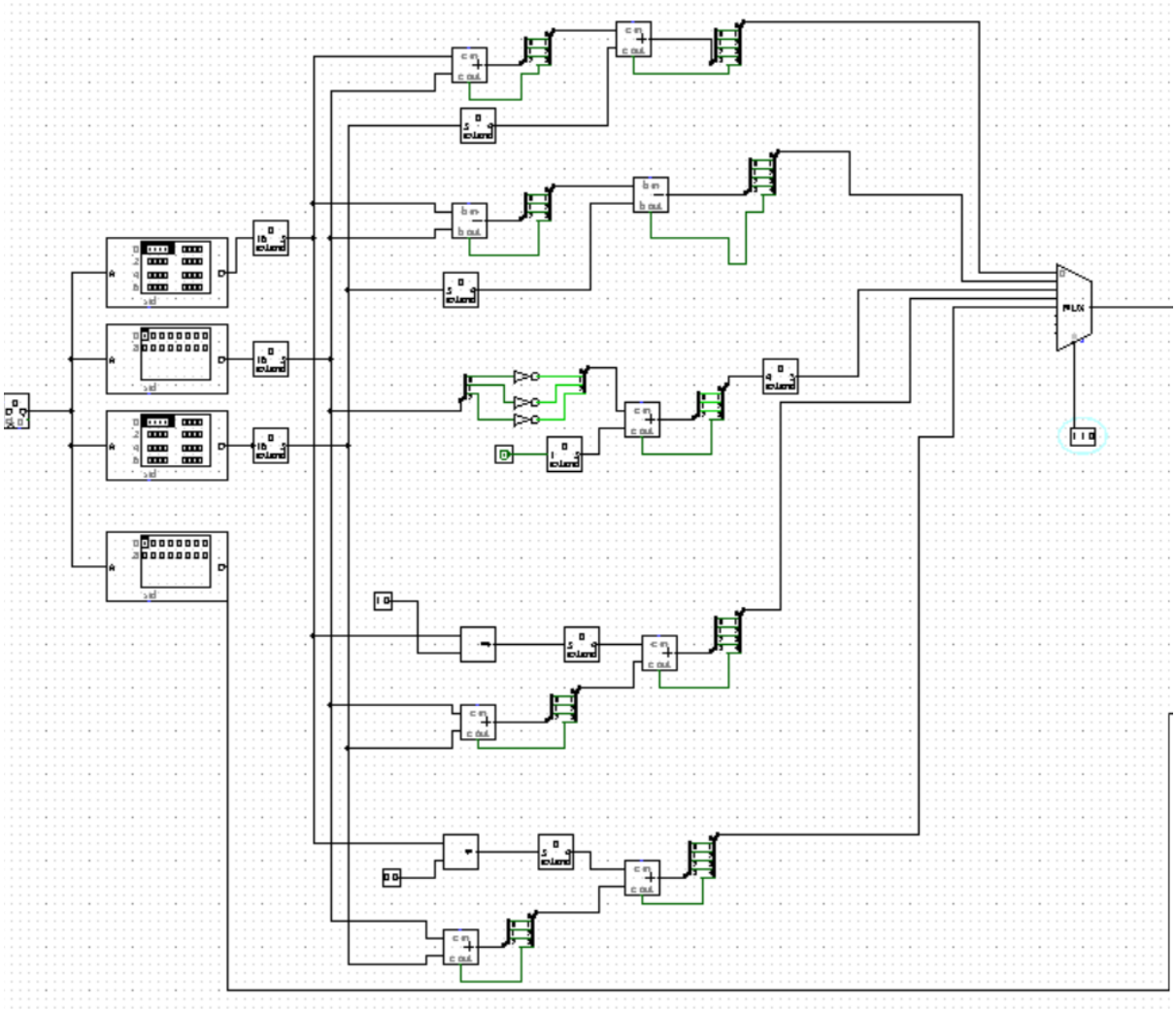
To create a general purpose computer, at first a 4-bit counter was taken:



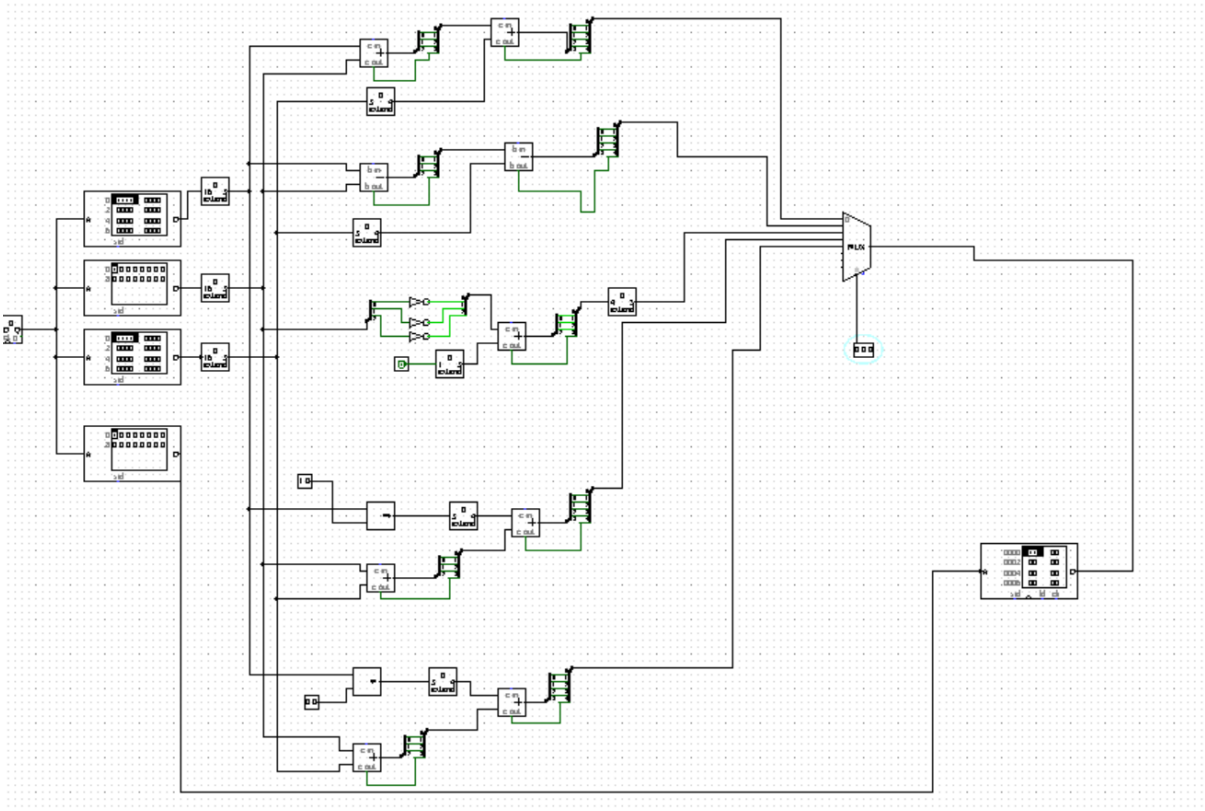
After taking counter, it was joined with 4 ROMs:



The counter bit and 3 ROMs were then joined to previously built ALU



After that , the output of ALU and the fourth ROM was joined to RAM



In this way, a general-purpose computer was developed by using the ALU previously built.

Logisim zip-file:

Logisim.zip