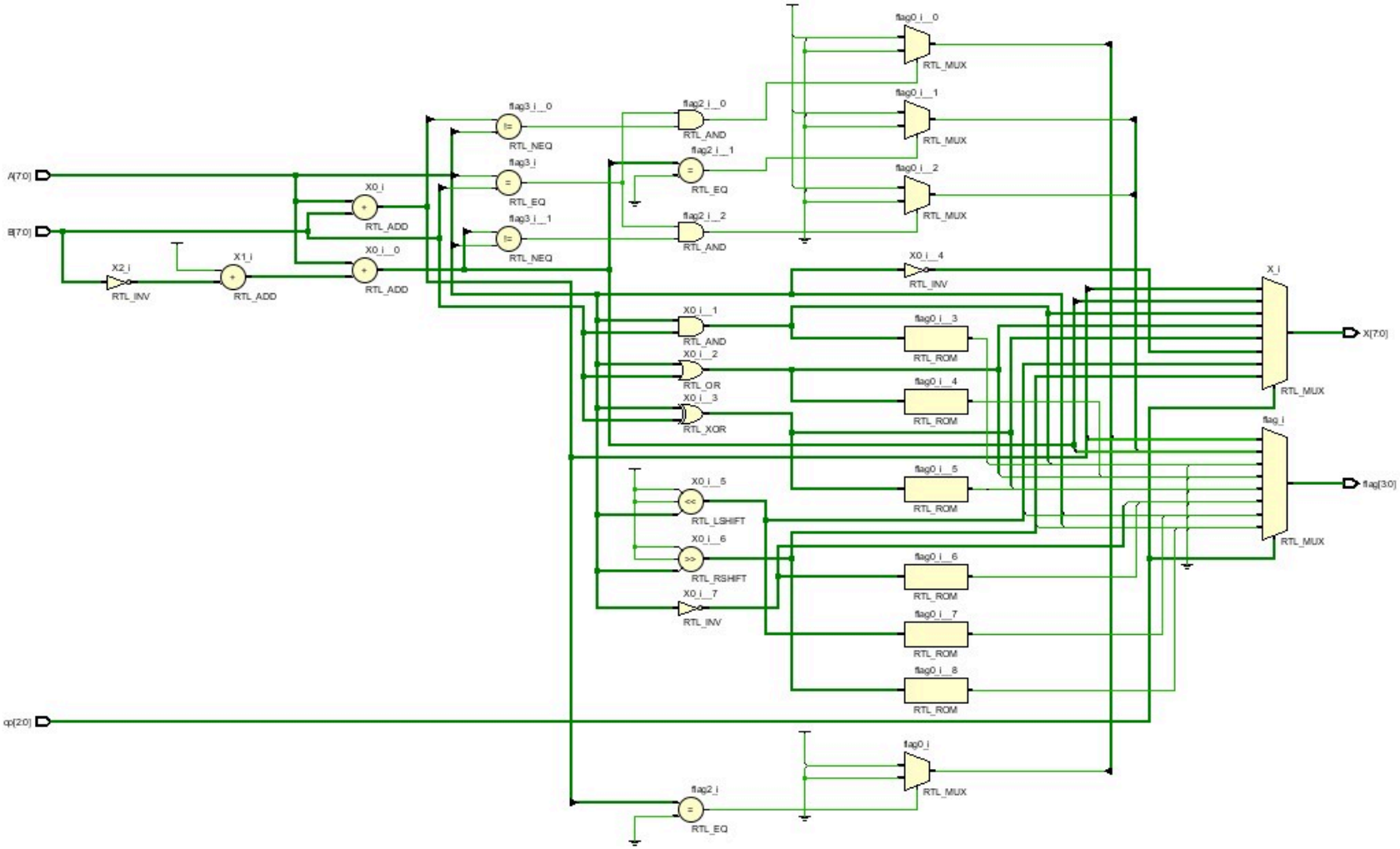
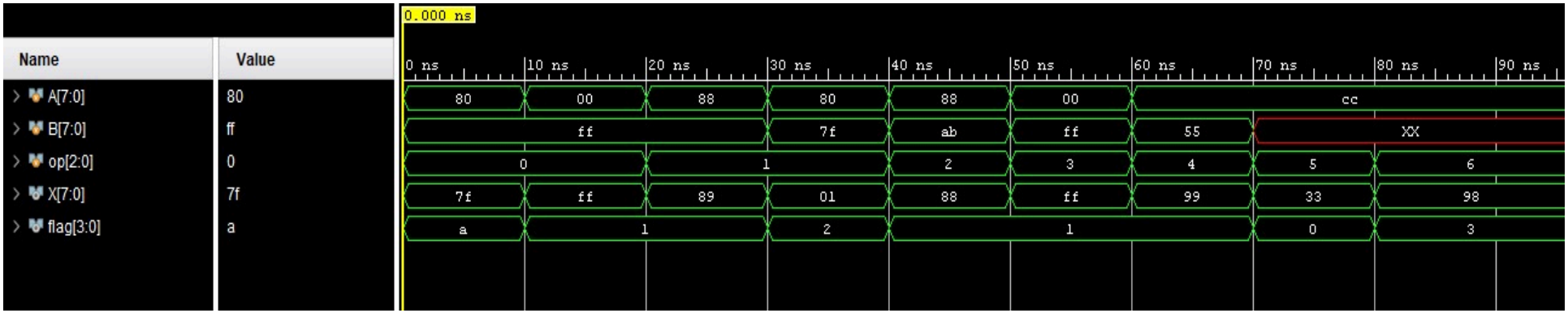
```
23      module ALU(
24          input [7:0] A,
25          input [7:0] B,
26          input [2:0] op,
27          output reg [7:0] X,
28          output reg [3:0] flag
29          );
30          reg [7:0] temp;
31          always@(A or B or op)
32          begin
33          flag[3] = 0;
34          flag[1] = 0;
35          case(op)
36
37          3'b000:
38          begin
39          {flag[1],X} = A+B;
40          flag[0] = X[7];
41          flag[2] = (X==0)?1:0;
42          flag[3]= ((A[7] == B[7]) & (X[7] != A[7])) ? 1 : 0;
43          end
44
45          3'b001:
46          begin
47          temp = ~B +1;
48          {flag[1],X} = A+temp;
49          flag[0] = X[7];
50          flag[2] = (X==0)?1:0;
51          flag[3]= ((A[7] == B[7]) & (X[7] != A[7])) ? 1 : 0;
52          end

54          3'b010:
55          begin
56          X = A & B ;
57          flag[0] = X[7];
58          flag[2] = (X==0)?1:0;
59          end
60
61          3'b011:
62          begin
63          X = A | B ;
64          flag[0] = X[7];
65          flag[2] = (X==0)?1:0;
66          end
67
68          3'b100:
69          begin
70          X = A ^ B ;
71          flag[0] = X[7];
72          flag[2] = (X==0)?1:0;
73          end
74
75          3'b101:
76          begin
77          X = ~A ;
78          flag[0] = X[7];
79          flag[2] = (X==0)?1:0;
80          end
81
82          3'b110:
83          begin
84          X = A << 1 ;
85          flag[1] = A[7];
86          flag[0] = X[7];
87          flag[2] = (X==0)?1:0;
88          end
89
90          3'b111:
91          begin
92          X = A >> 1;
93          flag[1] = A[0];
94          flag[0] = X[7];
95          flag[2] = (X==0)?1:0;
96          end
97
98          endcase
99
100         end
101     endmodule
```

```
23  module lab5_22_testbench;
24  reg [7:0] A;
25  reg [7:0] B;
26  reg [2:0] op;
27
28  wire [7:0] X;
29  wire [3:0] flag;
30
31  // Instantiate the Unit Under Test (UUT)
32  ALU uut (
33  .A(A),.B(B),.op(op),.X(X),.flag(flag)
34  );
35  initial begin
36  // Initialize Inputs
37  A=8'b10000000;
38  B=8'b11111111;
39  op=3'b000;
40  // Wait 100 ns for global reset to finish
41  #10;
42  A=8'b00000000;
43  B=8'b11111111;
44  op=3'b000;
45  // Wait 100 ns for global reset to finish
46  #10;
47  A=8'b10001000;
48  B=8'b11111111;
49  op=3'b001;
50  // Wait 100 ns for global reset to finish
51  #10;
52  A=8'b10000000;
53  B=8'b01111111;
54  op=3'b001;
55  // Wait 100 ns for global reset to finish
56  #10;
57  A=8'b10001000;
58  B=8'b10101011;
59  op=3'b010;
60  // Wait 100 ns for global reset to finish
61  #10;
62  A=8'b00000000;
63  B=8'b11111111;
64  op=3'b011;
65  // Wait 100 ns for global reset to finish
66  #10;
67  A=8'b11001100;
68  B=8'b01010101;
69  op=3'b100;
70  // Wait 100 ns for global reset to finish
71  #10;
72  A=8'b11001100;
73  B=8'bxxxxxxxx;
74  op=3'b101;
75  // Wait 100 ns for global reset to finish
76  #10;
77  A=8'b11001100;
78  B=8'bxxxxxxxx;
79  op=3'b110;
80  // Wait 100 ns for global reset to finish
81  #10;
82  end
83  endmodule
84
```


A[7:0]
B[7:0]
op[2:0]
X[7:0]
flag[3:0]
X2_i
RTL_INV
X1_i
RTL_ADD
X0_i
RTL_ADD
X0_i__0
RTL_ADD
flag3_i__0
RTL_NEQ
flag3_i
RTL_EQ
flag3_i__1
RTL_NEQ
flag2_i__0
RTL_AND
flag2_i__1
RTL_EQ
flag2_i__2
RTL_AND
X0_i__4
RTL_INV
X0_i__1
RTL_AND
X0_i__2
RTL_OR
X0_i__3
RTL_XOR
X0_i__5
RTL_LSHIFT
X0_i__6
RTL_RSHIFT
X0_i__7
RTL_INV
flag2_i
RTL_EQ
flag0_i__0
RTL_MUX
flag0_i__1
RTL_MUX
flag0_i__2
RTL_MUX
flag0_i__3
RTL_ROM
flag0_i__4
RTL_ROM
flag0_i__5
RTL_ROM
flag0_i__6
RTL_ROM
flag0_i__7
RTL_ROM
flag0_i__8
RTL_ROM
flag0_i
RTL_MUX
X_i
RTL_MUX
flag_i
RTL_MUX


0.000 ns
Name
Value
A[7:0]
80
B[7:0]
ff
op[2:0]
0
X[7:0]
7f
flag[3:0]
a
0 ns
10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
70 ns
80 ns
90 ns
80
00
88
80
88
00
cc
ff
7f
ab
ff
55
XX
0
1
2
3
4
5
6
7f
ff
89
01
88
ff
99
33
98
a
1
2
1
0
3