Assignment Submission Sheet

Term: 321221 **Submission Date: 22-11-2021**

Assignment Number: 05

Course Code: ECE290 **Section: E1901** **Group: A**

Registration Number: 11904463 **Student Name: Mohit Rawat** **Roll No: 09**

1. **Concept Learned**

   I have learn about how to make 16 x 1 Multiplexer using 4 x 1 multiplexer and how to code in Verilog and make simulation of it.

2. **Key Observations &Insights**

   Key observation is the output waveform was the multiplexer's output according to select line and the input given throw testbench.

3. **Application Areas**

   Application of Multiplexer is in modern devices like microprocessor inside ALU.

4. A Verilog Program to implement 16 x 1 Multiplexer using 4 x 1 Multiplexers.

Date.
Page.

4x1 Multiplexer.

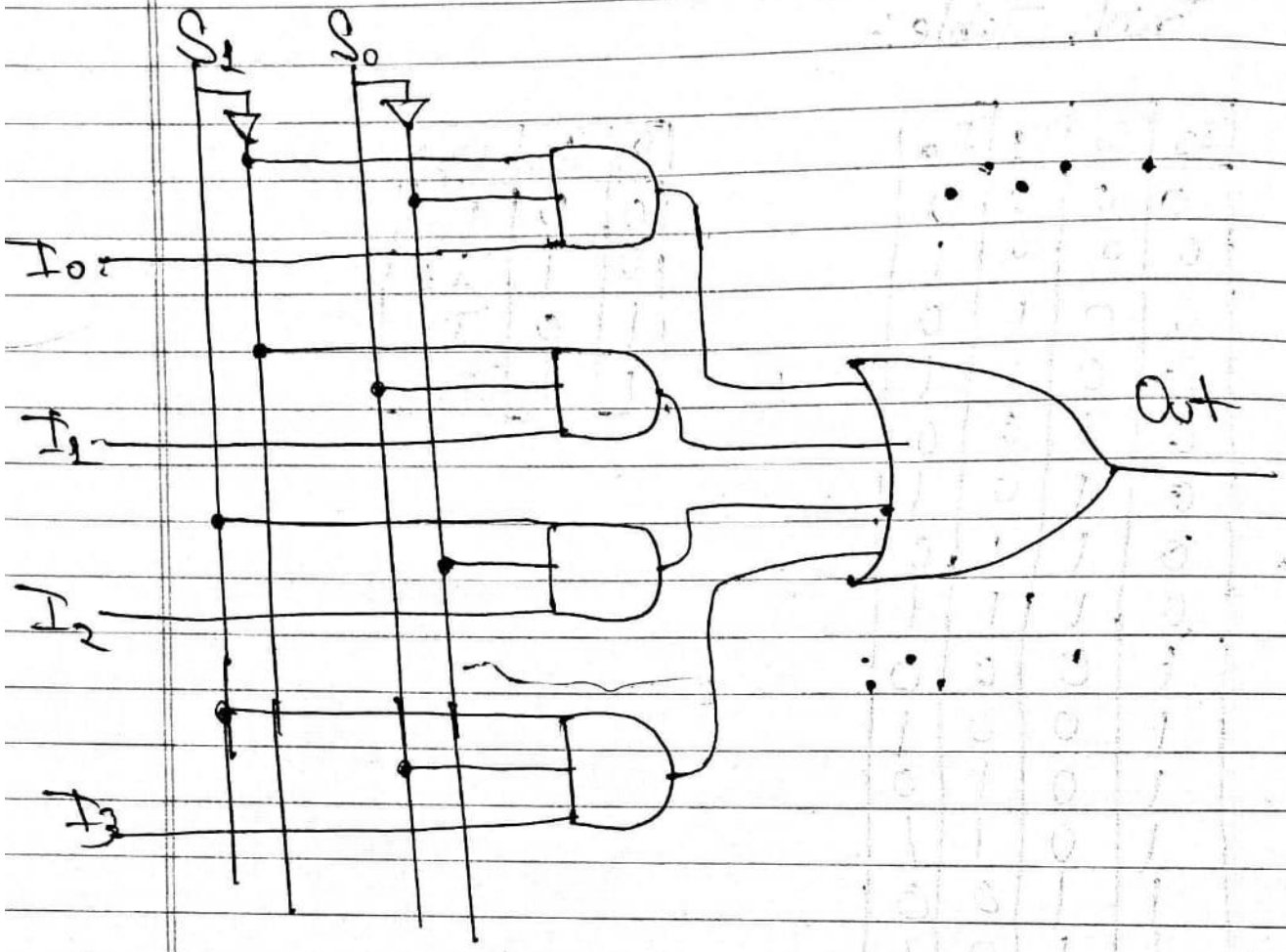
Truth table:-

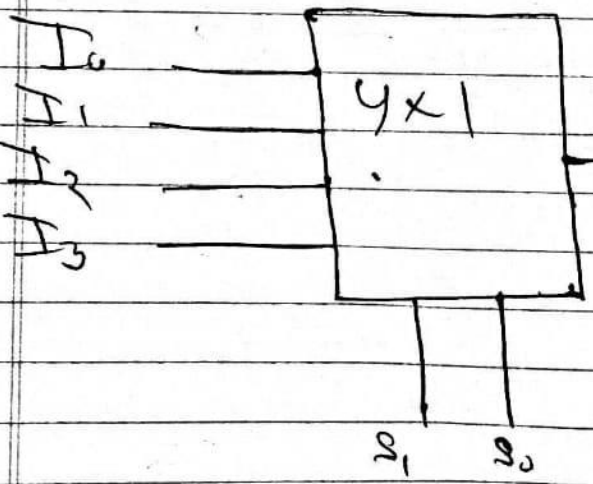
I_3	I_2	I_1	I_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

S_1	S_0	Out
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Date.
Page.

Circuit Diagram of 4 x 1 Mux.





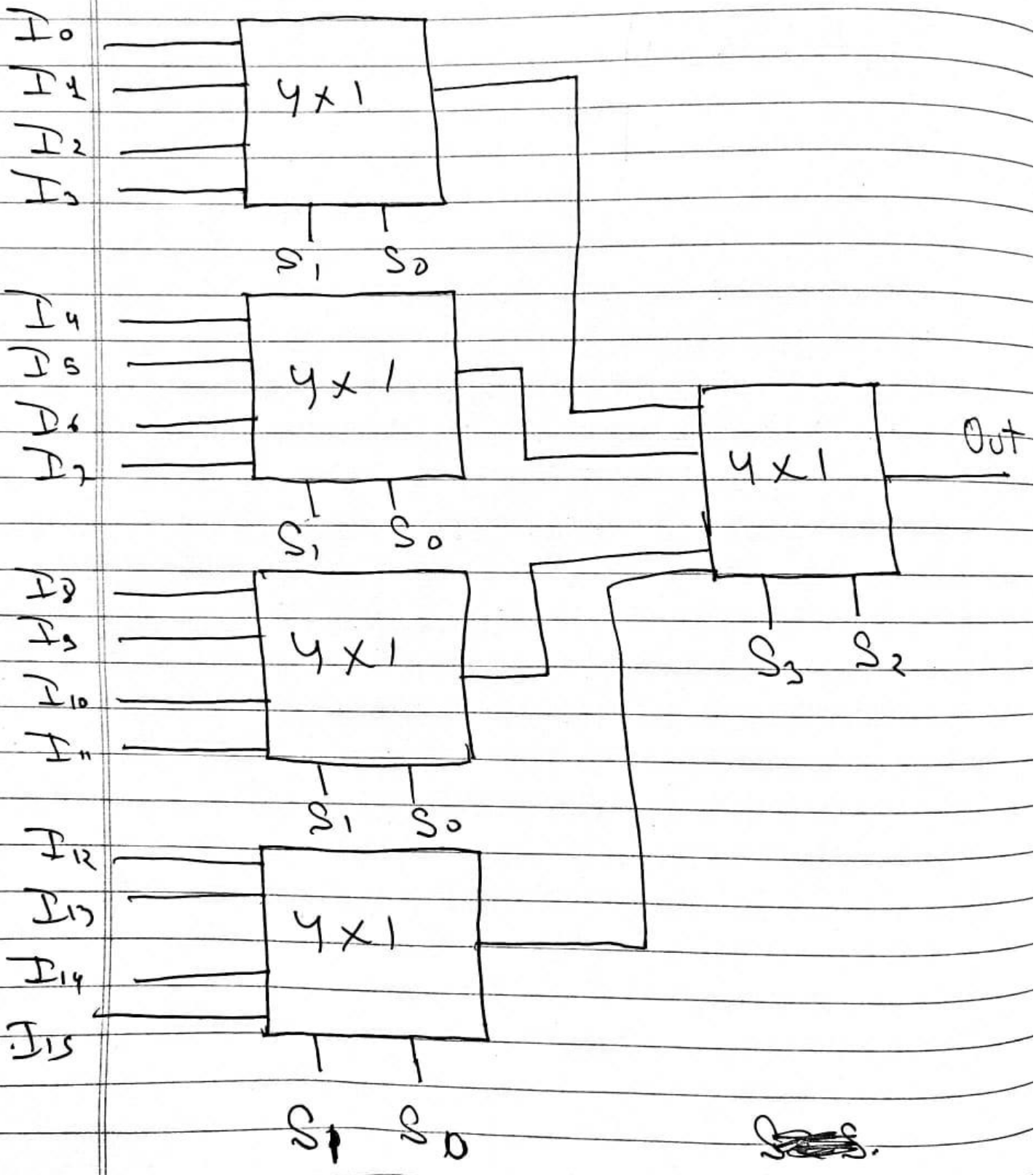
16 X 1 Mutiplexer using 4 X 1 Mutiplexer

Truth Table.

S_3	S_2	S_1	S_0	Out
0	0	0	0	I_0
0	0	0	1	I_1
0	0	1	0	I_2
0	0	1	1	I_3
0	1	0	0	I_4
0	1	0	1	I_5
0	1	1	0	I_6
0	1	1	1	I_7
1	0	0	0	I_8
1	0	0	1	I_9
1	0	1	0	I_{10}
1	0	1	1	I_{11}
1	1	0	0	I_{12}
1	1	0	1	I_{13}
1	1	1	0	I_{14}
1	1	1	1	I_{15}

Date.
Page.

Circuit Diagram 16×1



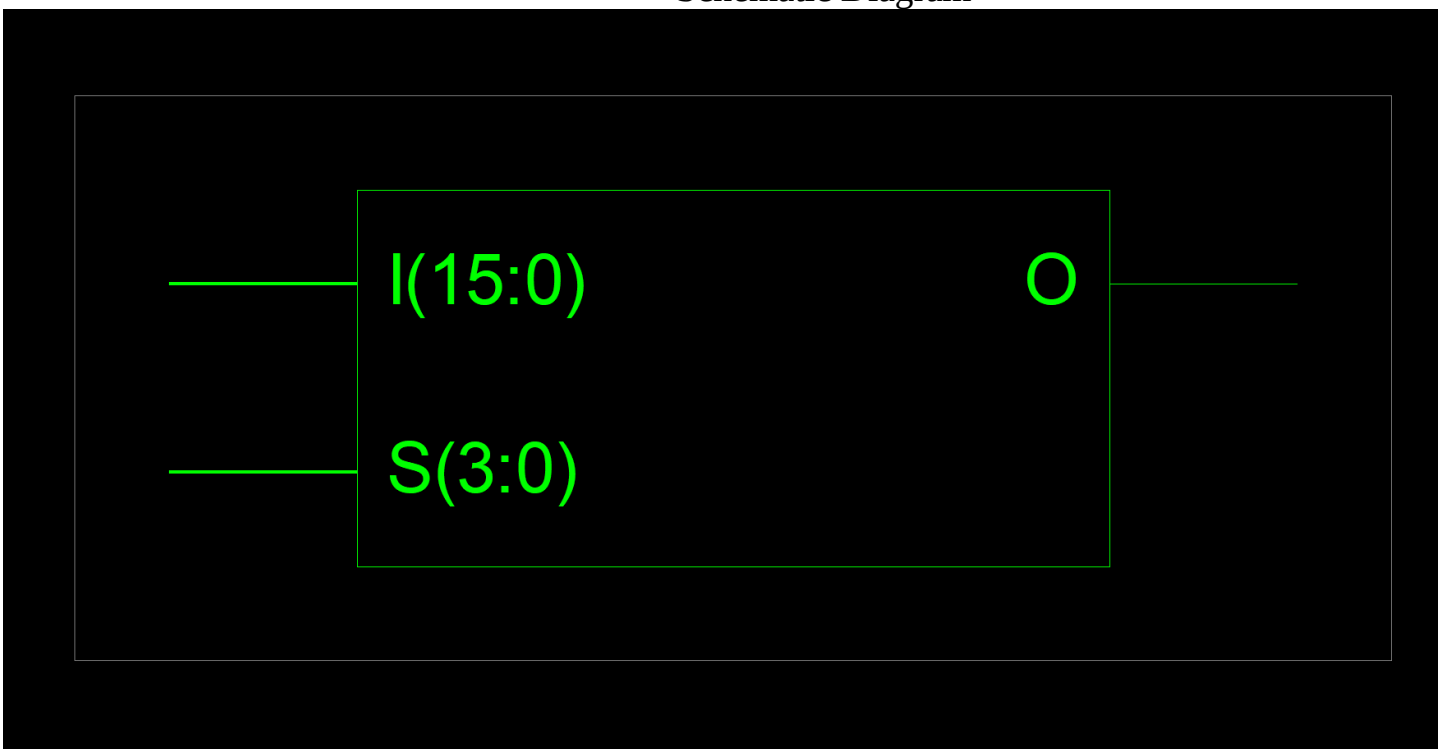
Verilog Code

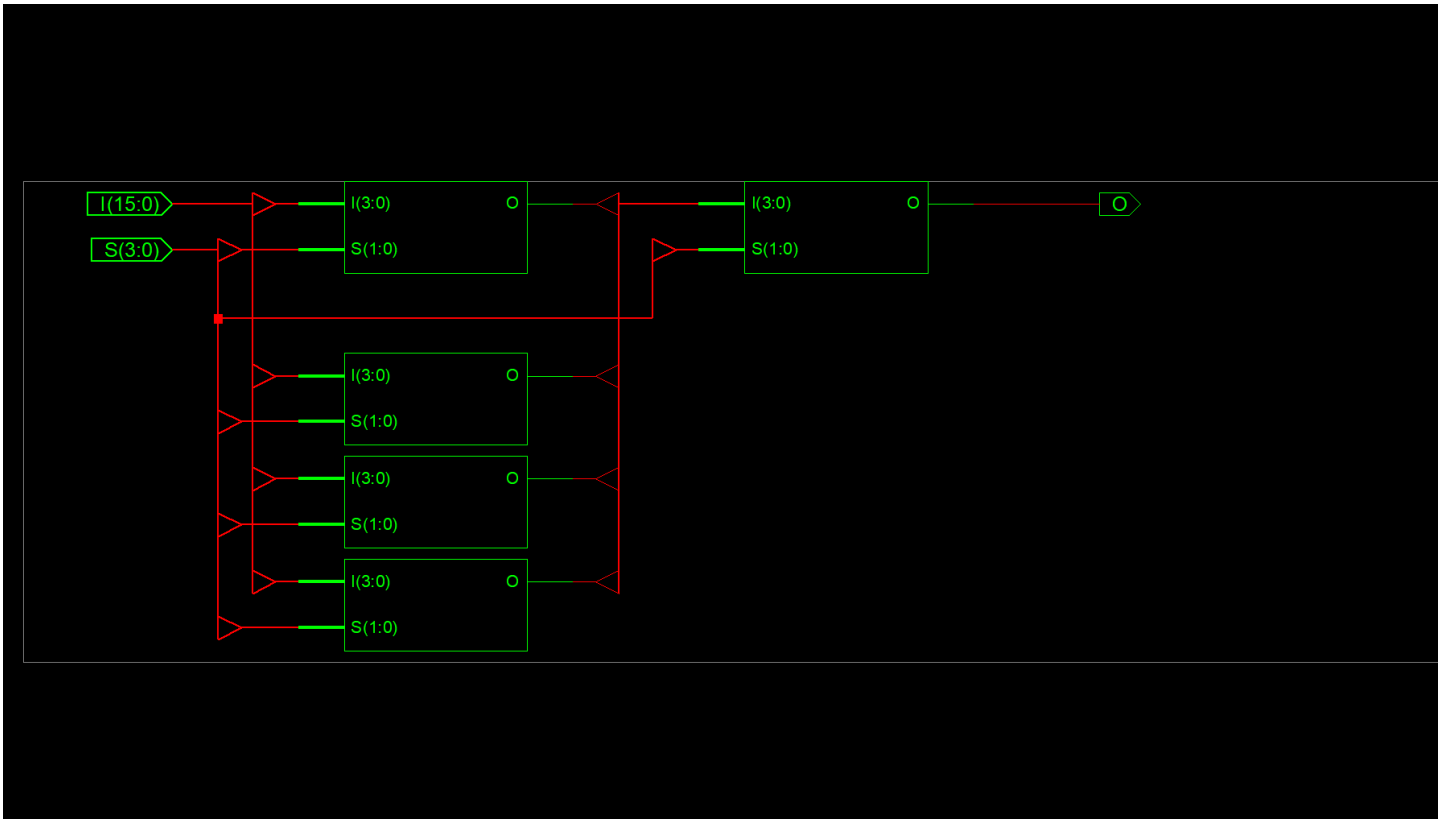
```
1  `timescale 1ns / 1ps
2
3  module Mux_4x1(I, S, O);
4  input I;
5  wire [3:0] I;
6  input S;
7  wire [1:0] S;
8
9  output O;
10 reg O;
11
12
13 always @ (I or S)
14 begin
15    if(S == 2'b00)
16       O <=I[0];
17    else if(S == 2'b01)
18       O <= I[1];
19    else if(S == 2'b10)
20       O <= I[2];
21    else if(S == 2'b11)
22       O<= I[3];
23 end
24
25 endmodule
26
27
28 module Mux_16x1(I, S, O);
29 input [15:0] I;
30 input [3:0] S;
31 output O;
32
33 wire [3:0] w;
34
35 Mux_4x1 M4_1(I[3:0],   S[1:0], w[0]);
36 Mux_4x1 M4_2(I[7:4],   S[1:0], w[1]);
37 Mux_4x1 M4_3(I[11:8],  S[1:0], w[2]);
38 Mux_4x1 M4_4(I[15:12], S[1:0], w[3]);
39
40 Mux_4x1 M4_5(w, S[3:2], O);
41
42 endmodule
43
```

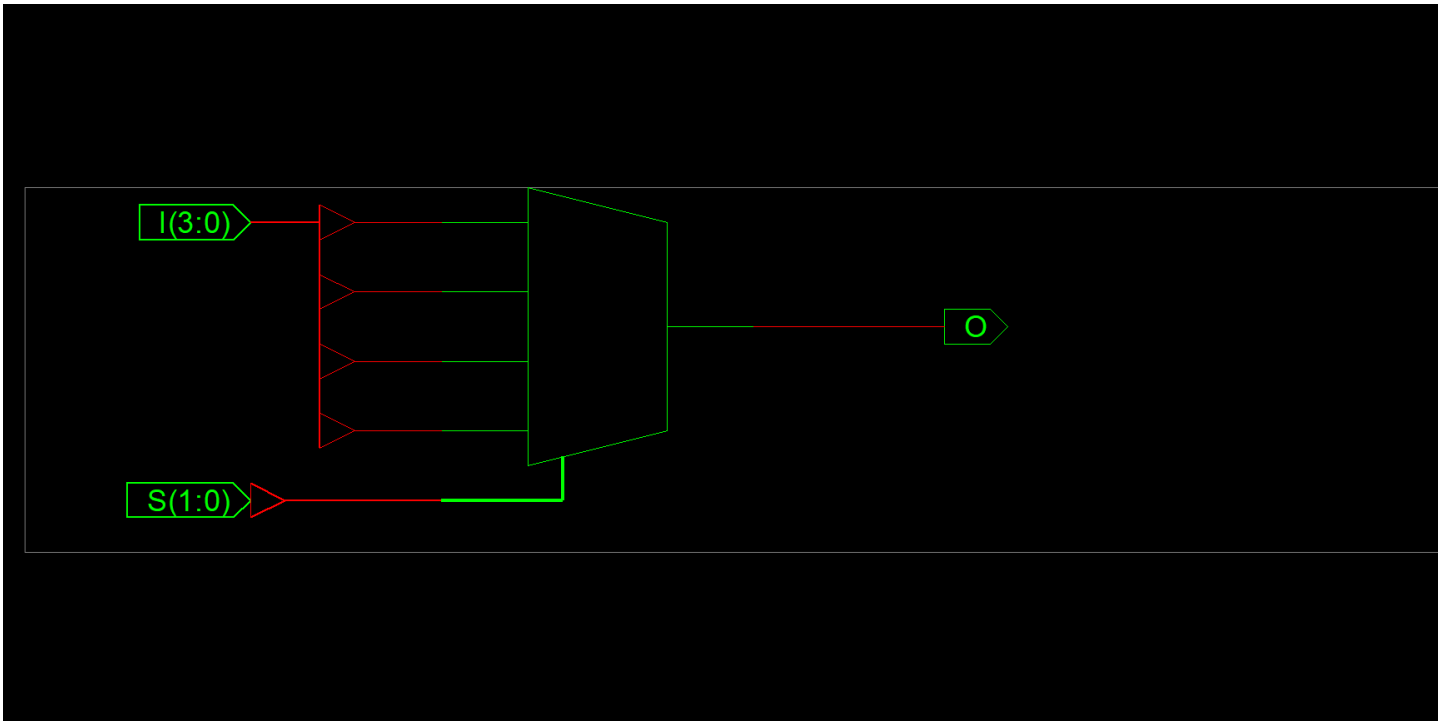
Test Bench For Verilog Code

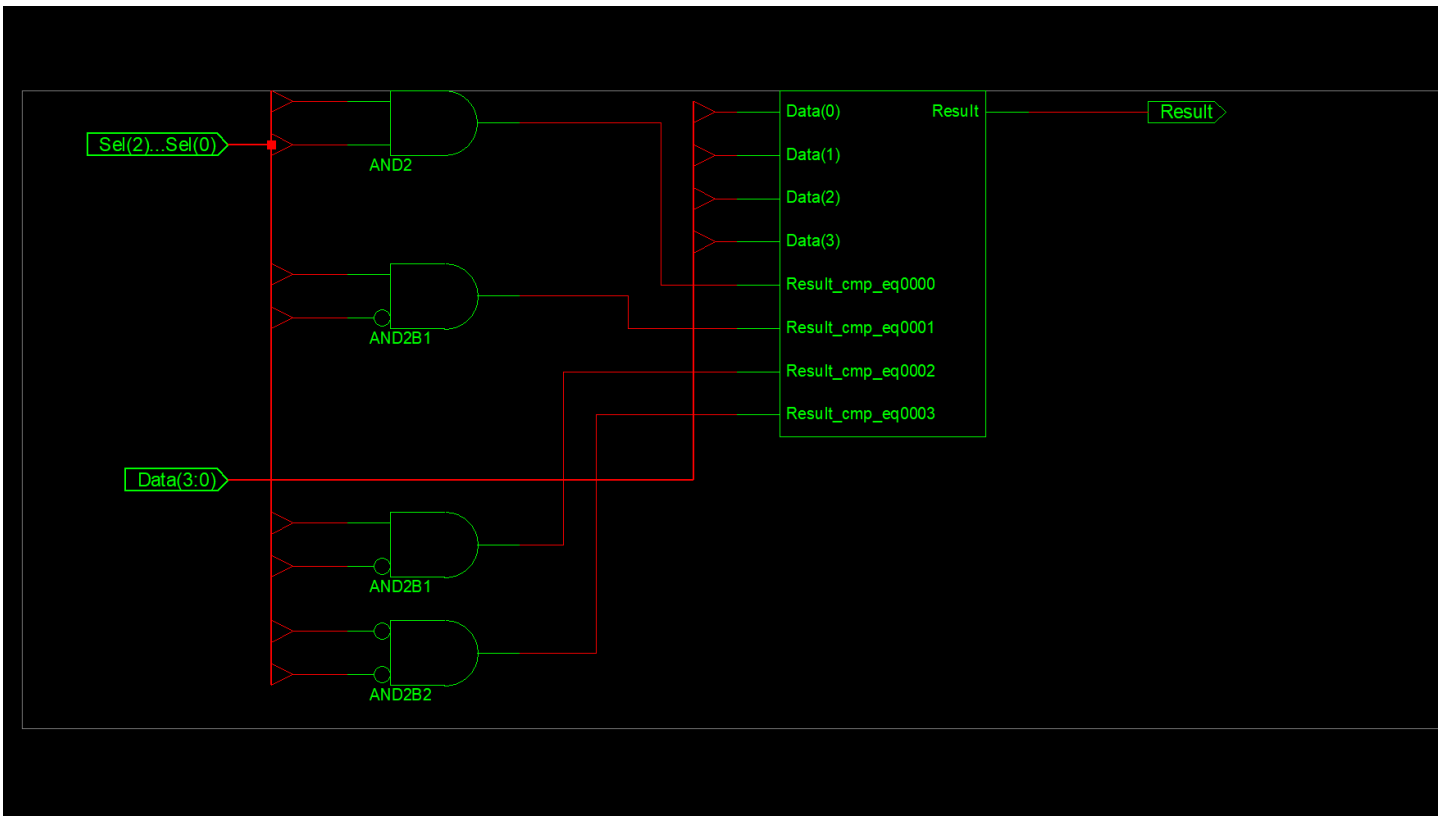
```
`timescale 1ns / 1ps


module Mux_16x1_TB_v;

   // Inputs
   reg [15:0] I;
   reg [3:0] S;

   // Outputs
   wire O;

   // Instantiate the Unit Under Test (UUT)
   Mux_16x1 uut (
      .I(I),
      .S(S),
      .O(O)
   );

   initial begin
      // Initialize Inputs
      I = 0;
      S = 0;
      // Wait 100 ns for global reset to finish
      #5;

      I=16'b0000000000000001; S=4'b0000; #10;
      I=16'b0000000000000010; S=4'b0001; #10;
      I=16'b0000000000000100; S=4'b0010; #10;
      I=16'b0000000000001000; S=4'b0011; #10;
      I=16'b0000000000010000; S=4'b0100; #10;
      I=16'b0000000000100000; S=4'b0101; #10;
      I=16'b0000000001000000; S=4'b0110; #10;
      I=16'b0000000010000000; S=4'b0111; #10;
      I=16'b0000000100000000; S=4'b1000; #10;
      I=16'b0000001000000000; S=4'b1001; #10;
      I=16'b0000010000000000; S=4'b1010; #10;
      I=16'b0000100000000000; S=4'b1011; #10;
      I=16'b0001000000000000; S=4'b1100; #10;
      I=16'b0010000000000000; S=4'b1101; #10;
      I=16'b0100000000000000; S=4'b1110; #10;
      I=16'b1000000000000000; S=4'b1111; #10;
   end
endmodule
```

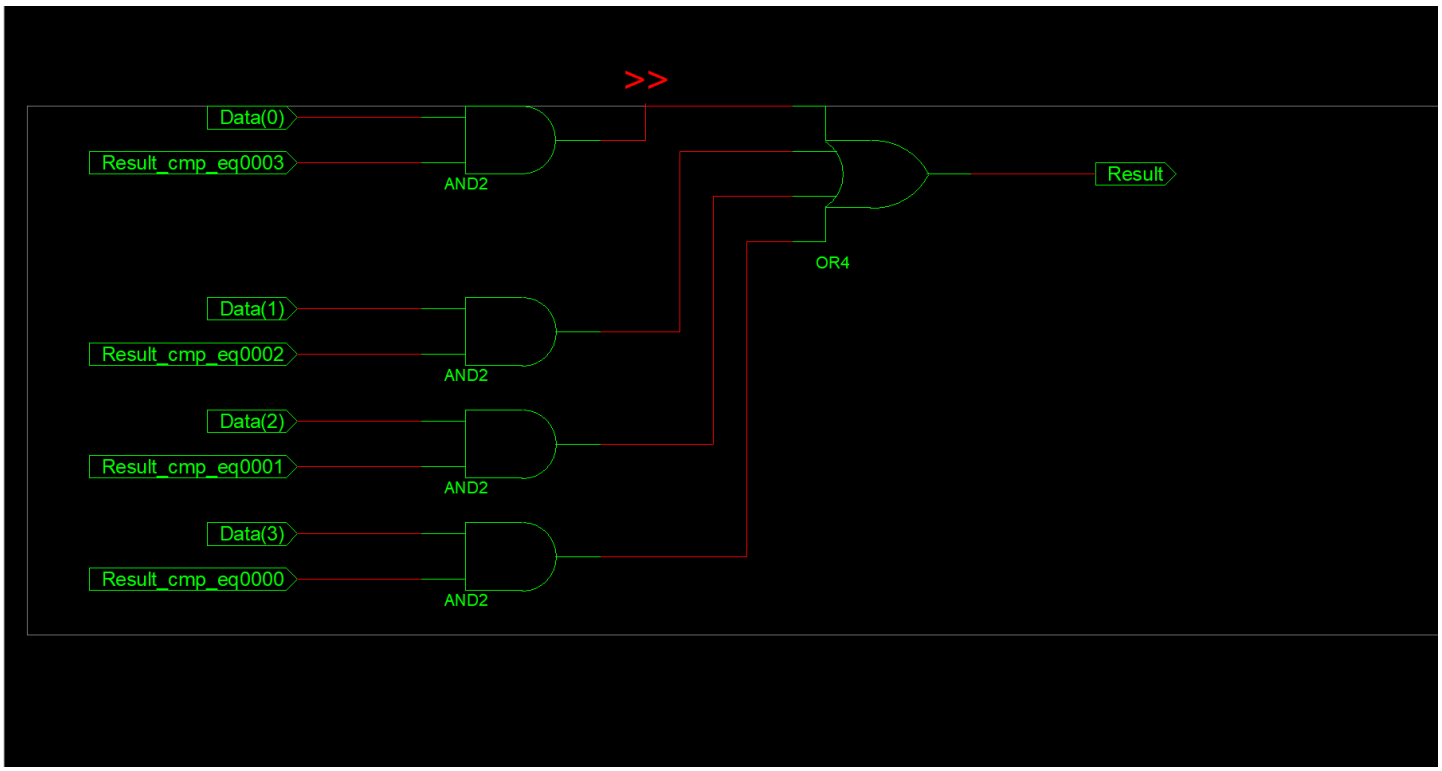
Schematic Diagram



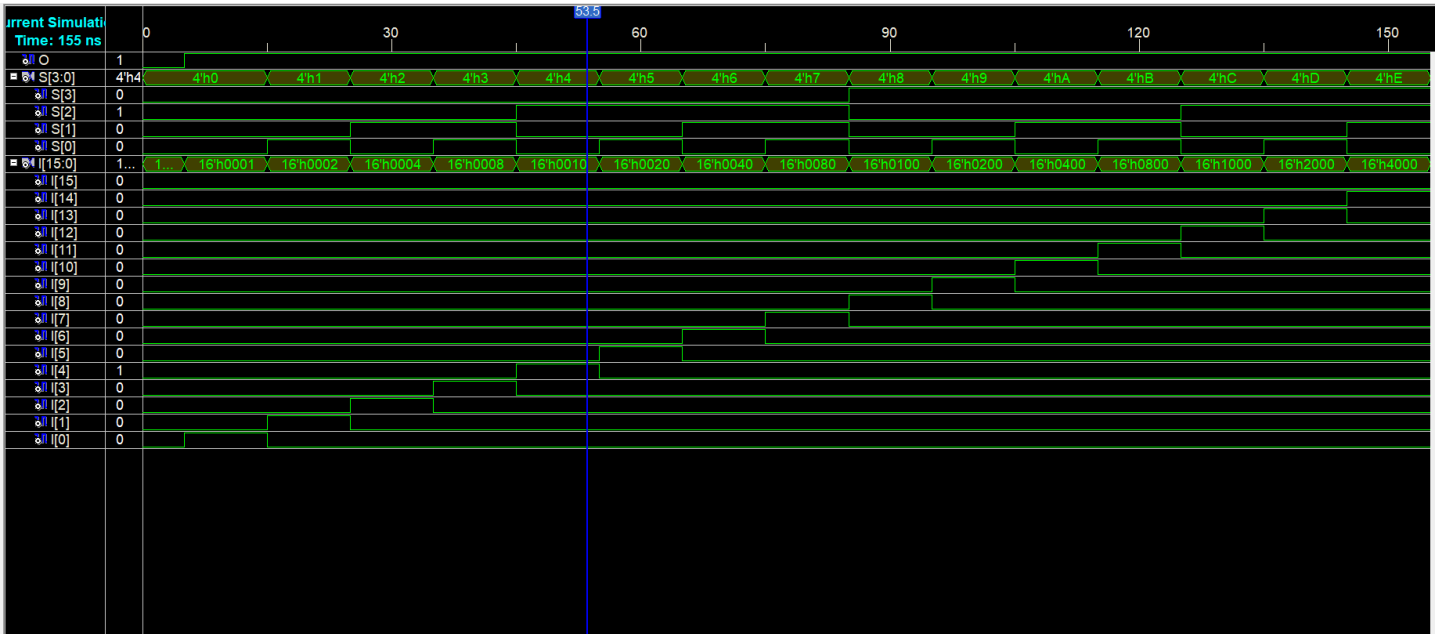



I(3:0)
O
S(1:0)


Sel(2)...Sel(0)
AND2
AND2B1
Data(3:0)
AND2B1
AND2B2
Data(0)
Result
Data(1)
Data(2)
Data(3)
Result_cmp_eq0000
Result_cmp_eq0001
Result_cmp_eq0002
Result_cmp_eq0003
Result



Wave Form



END