Assignment Submission Sheet

Term: 321221	**Submission Date: 05-10-2021**	
	Assignment Number: 03	
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 2x1, 4x1 and 16x1 (using 4x1 and 2x1) Multiplexer in verilog code and how to simulate its output in waveform.

2. **Key Observations &Insights**

   Key observation is the output waveform of 16x1 using 4x1 and 2x1 multiplexer and how we connect each other in verilog code.

3. **Application Areas**

   Multiplexer is used in many area like in processor to reduce number of pin.

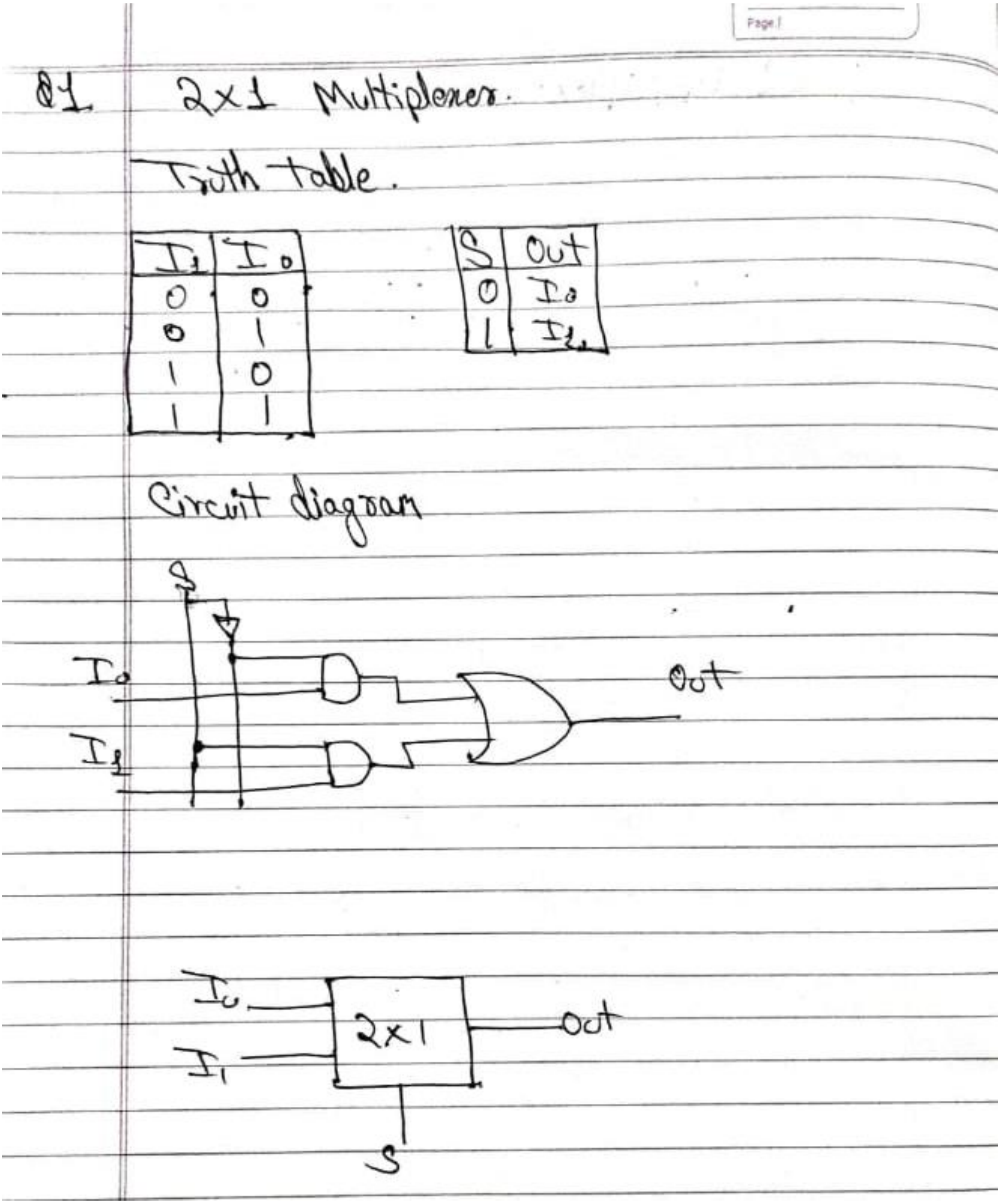
4. **A)**

   1. **2x1 Multiplexer using conditional operator.**
   2. **4x1 Multiplexer using conditional operator.**

   B)

   16x1 multiplexer using 2x1 and 4x1 multiplexer in data flow model.

2x1 Multiplexer



Page 1

Q1. 2x1 Multiplexer.

Truth table.

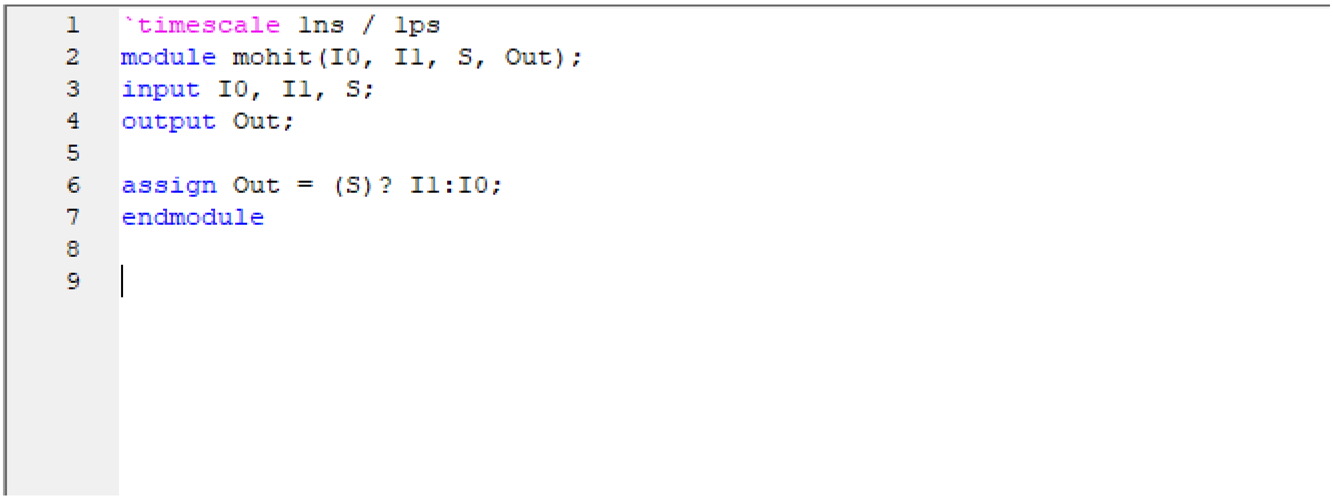
I_1	I_0
0	0
0	1
1	0
1	1

S	Out
0	I_0
1	I_1

Circuit diagram

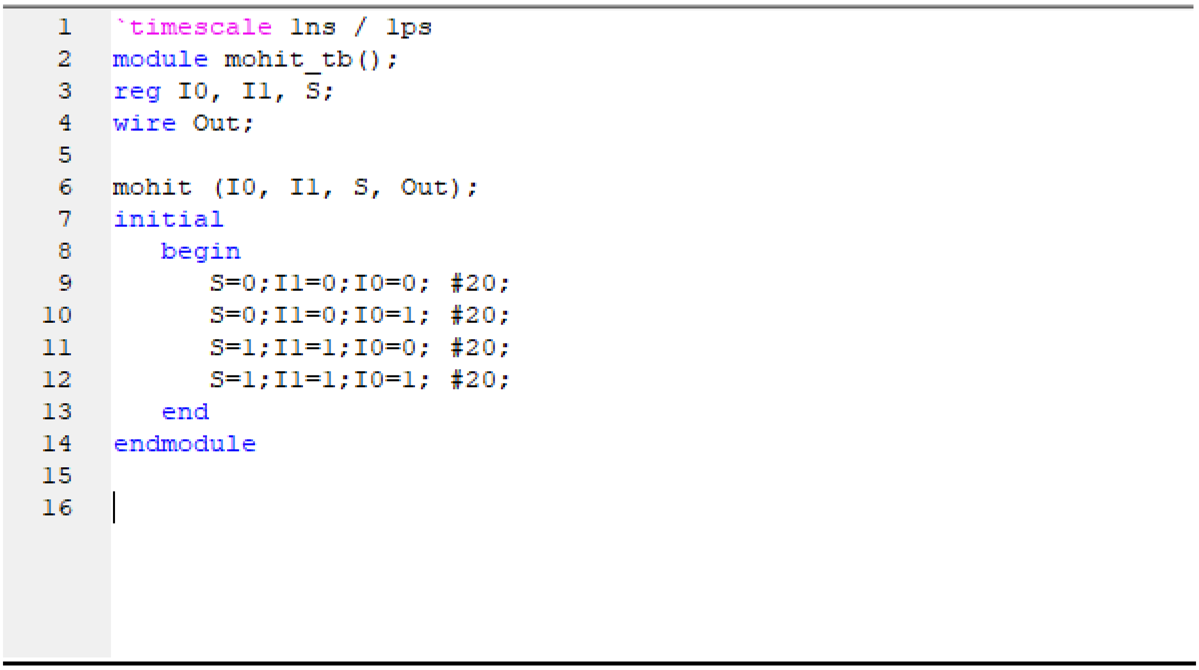
S
I_0
I_1
Out
I_0
I_1
2x1
Out
S

Verilog Code



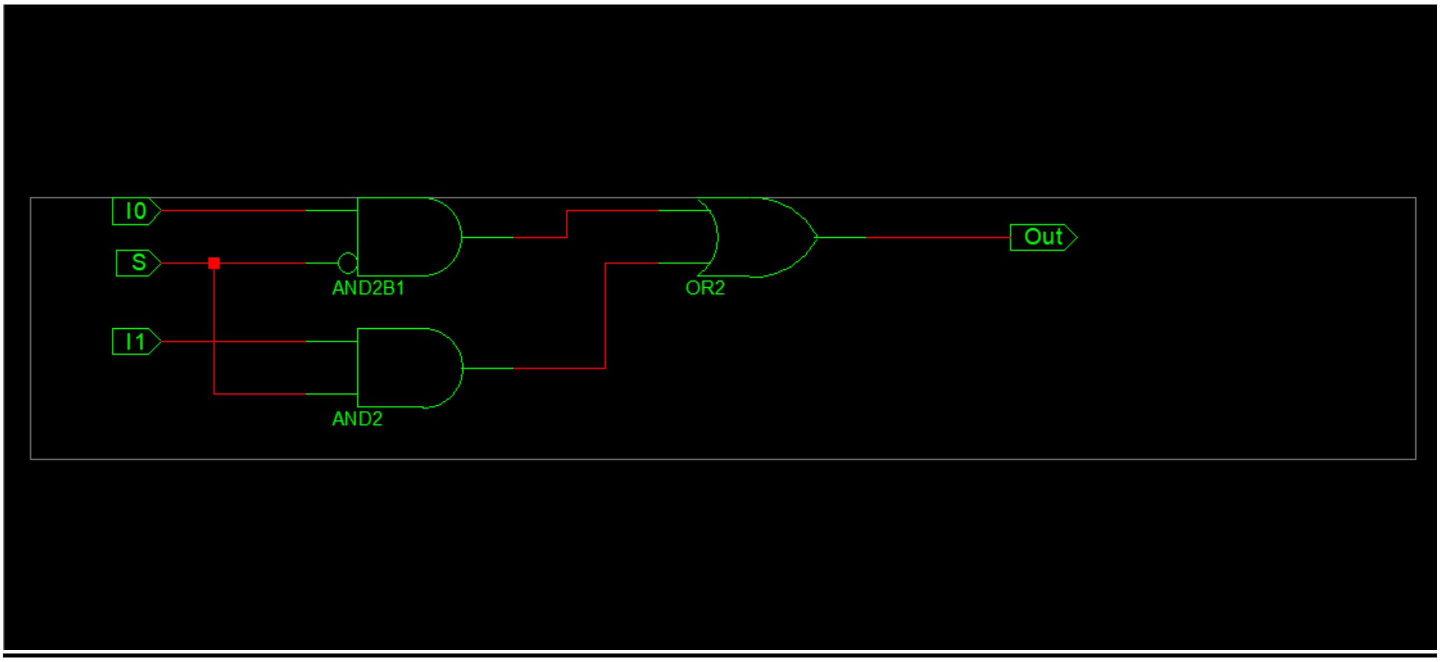
```
`timescale 1ns / 1ps
module mohit(I0, I1, S, Out);
input I0, I1, S;
output Out;

assign Out = (S)? I1:I0;
endmodule
```

Test bench code

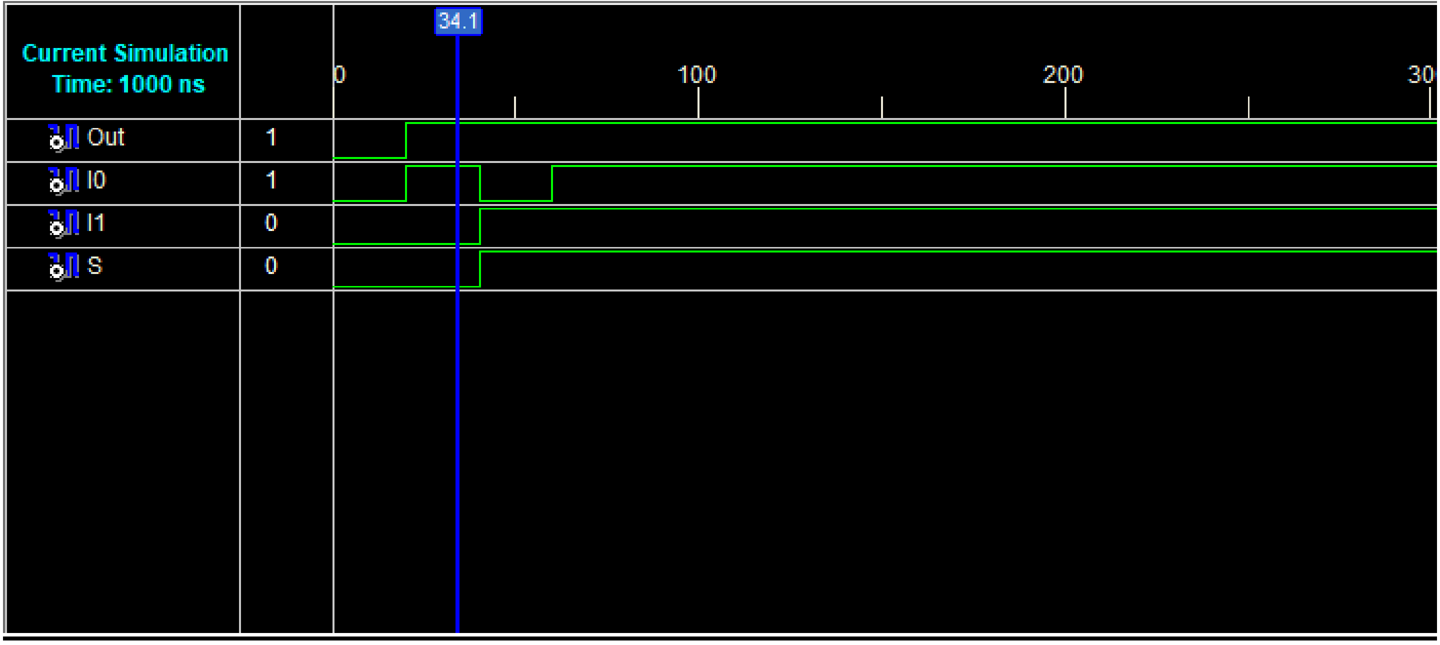


```
`timescale 1ns / 1ps
module mohit_tb();
reg I0, I1, S;
wire Out;

mohit (I0, I1, S, Out);
initial
   begin
      S=0;I1=0;I0=0; #20;
      S=0;I1=0;I0=1; #20;
      S=1;I1=1;I0=0; #20;
      S=1;I1=1;I0=1; #20;
   end
endmodule
```

Circuit design

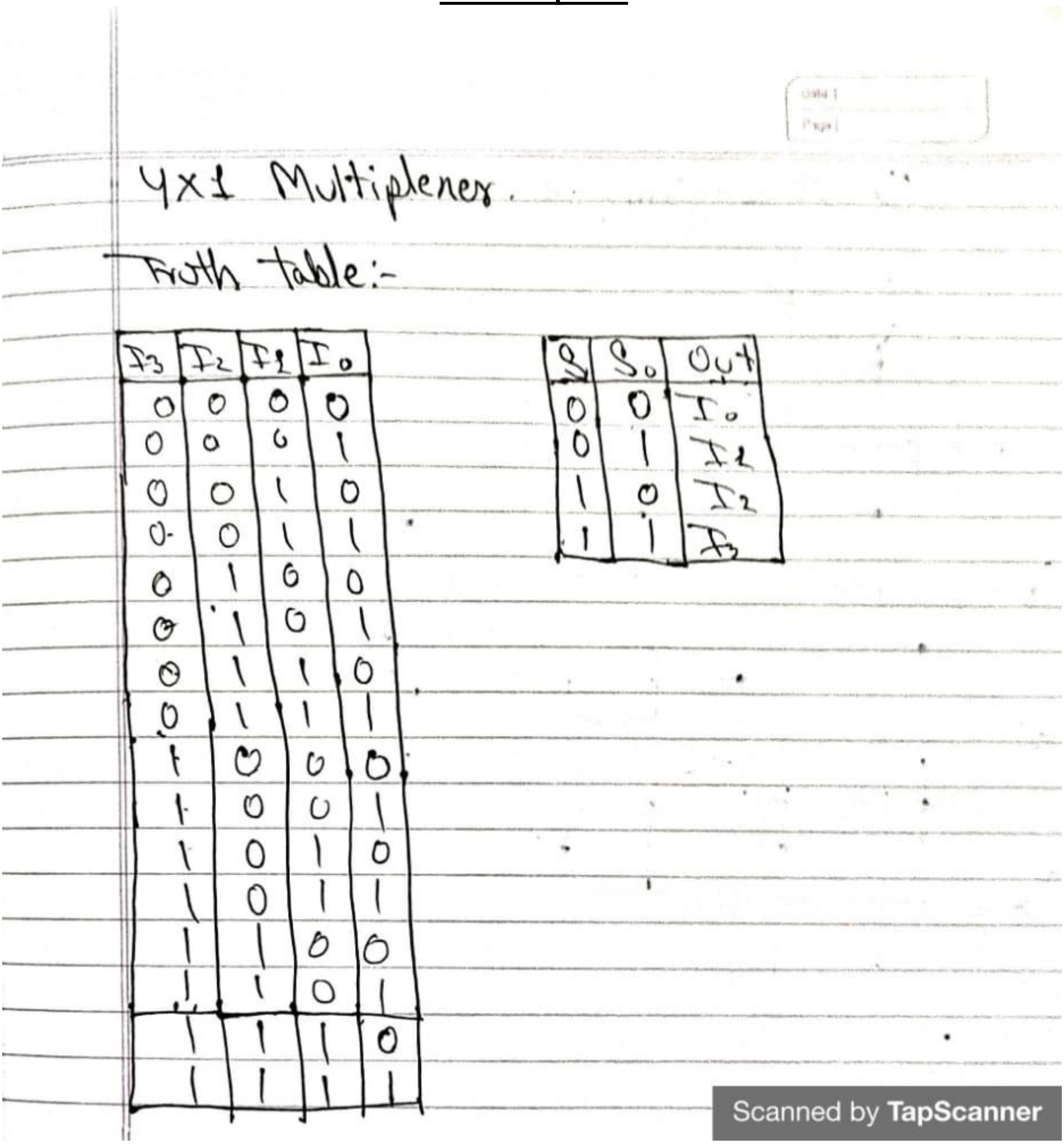


Wave form


Current Simulation Time: 1000 ns

Out	1
I0	1
I1	0
S	0

4x1 Multiplexer



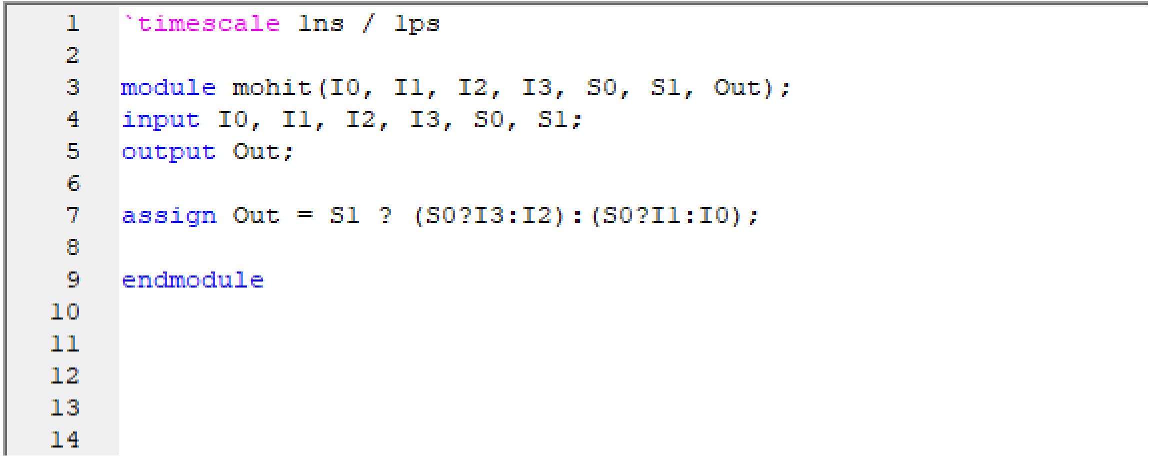
4x1 Multiplener.

Truth table:-

I3	I2	I1	I0
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

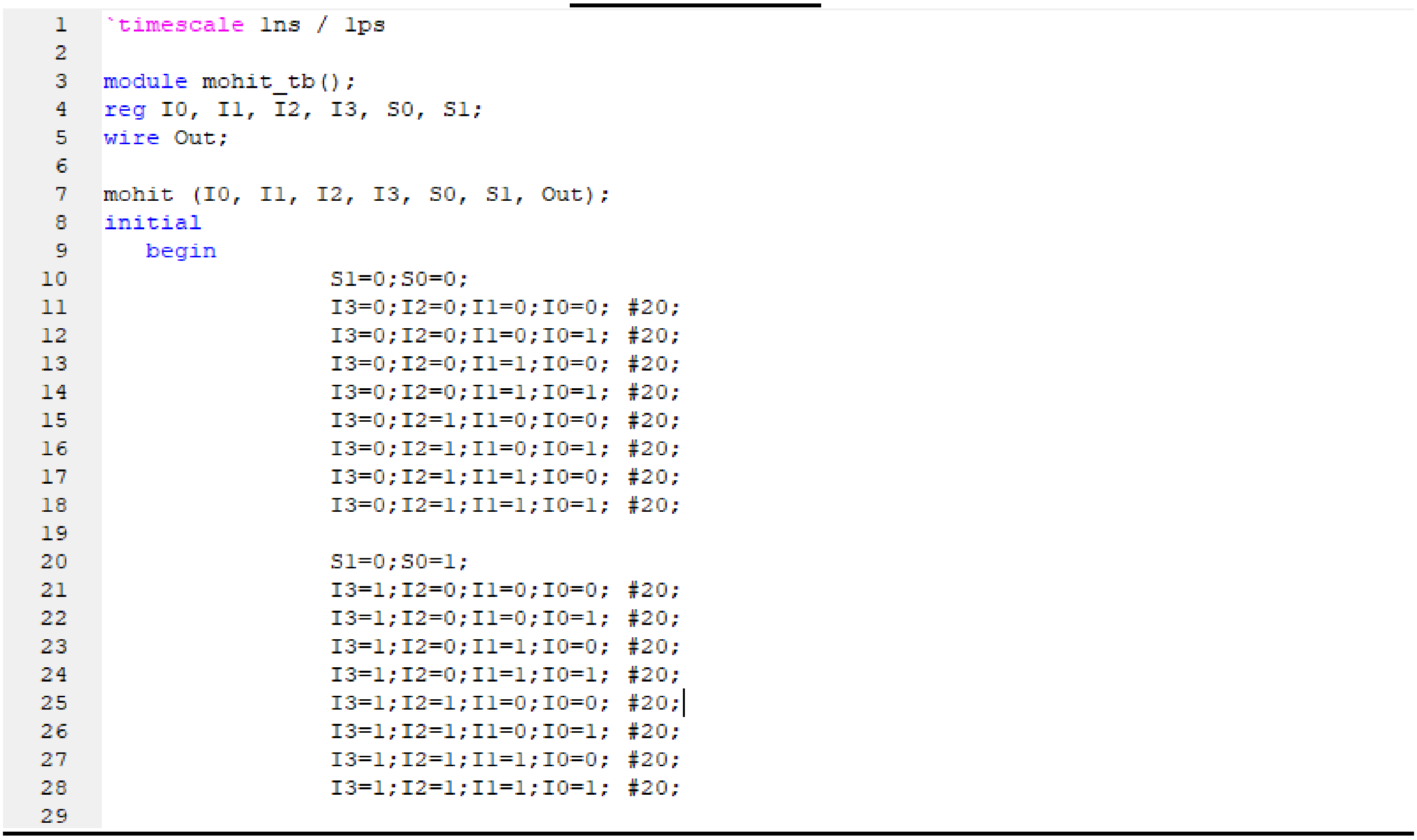
S_1	S_0	Out
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Verilog Code

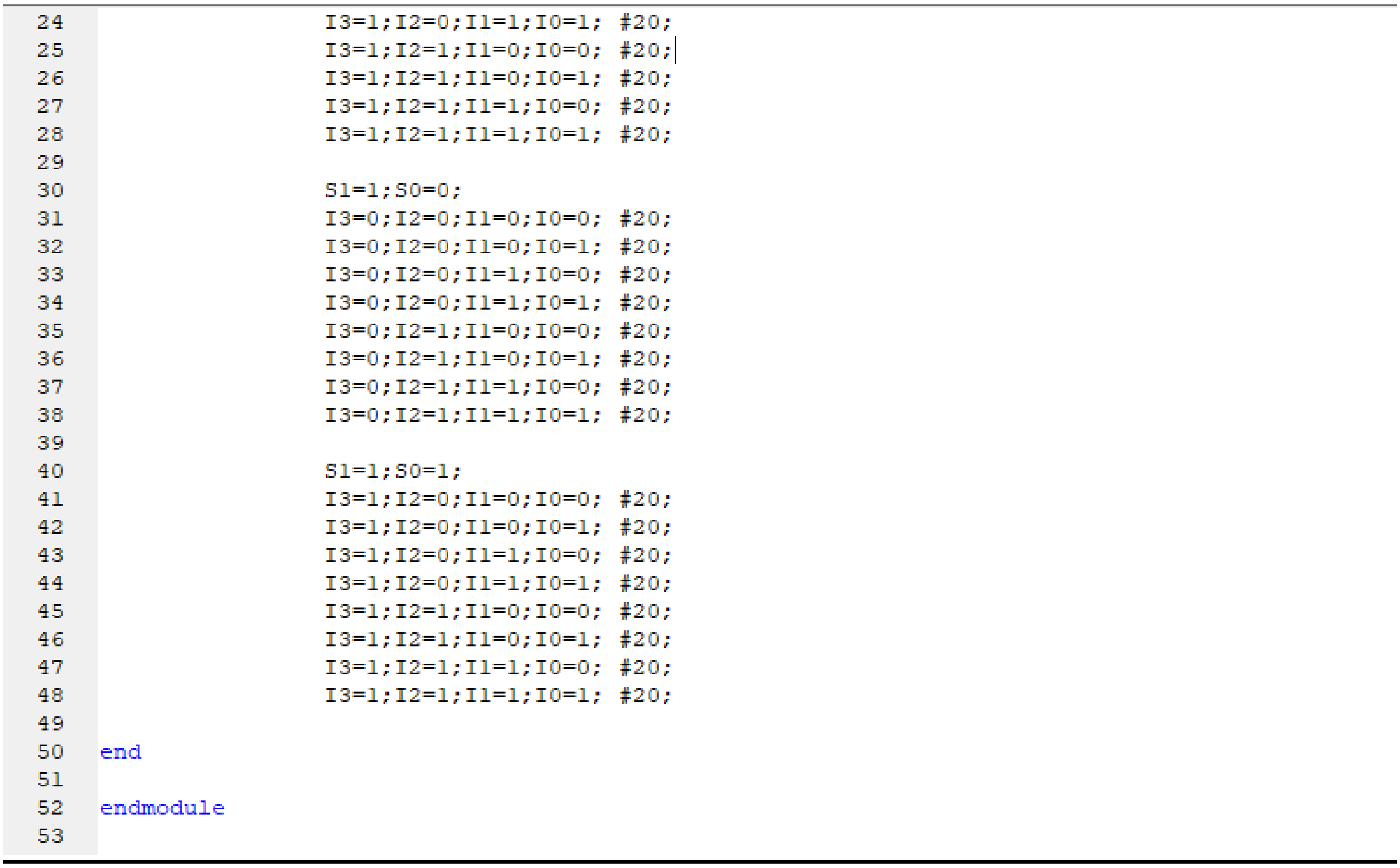


```
`timescale 1ns / 1ps

module mohit(I0, I1, I2, I3, S0, S1, Out);
input I0, I1, I2, I3, S0, S1;
output Out;

assign Out = S1 ? (S0?I3:I2):(S0?I1:I0);

endmodule
```

Test bench code

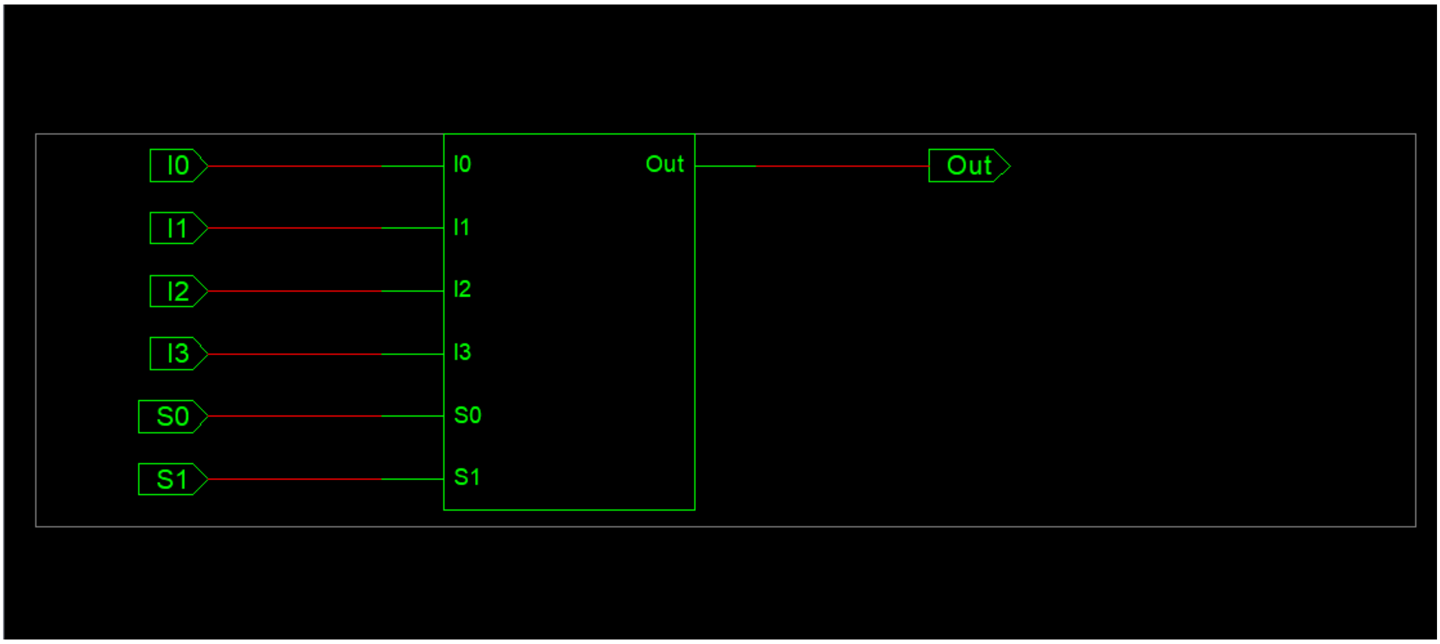


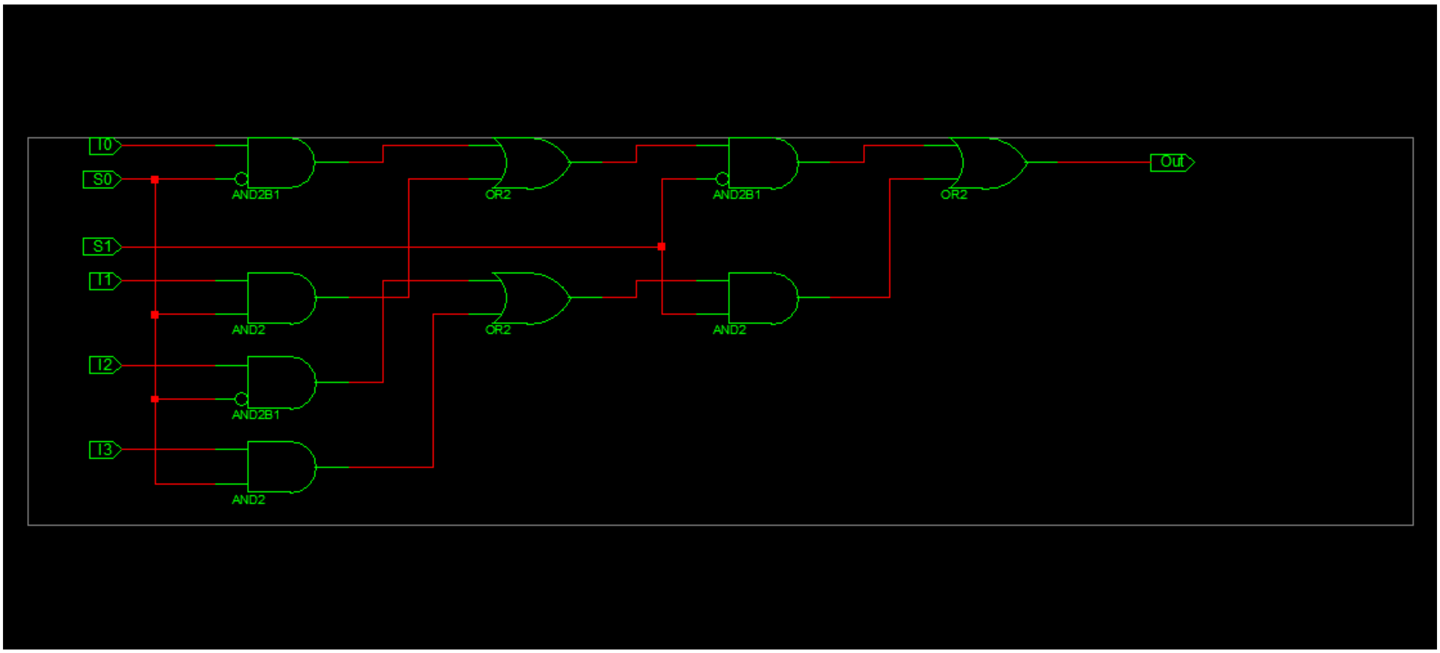
```
`timescale 1ns / 1ps

module mohit_tb();
reg I0, I1, I2, I3, S0, S1;
wire Out;

mohit (I0, I1, I2, I3, S0, S1, Out);
initial
   begin
                S1=0;S0=0;
                I3=0;I2=0;I1=0;I0=0; #20;
                I3=0;I2=0;I1=0;I0=1; #20;
                I3=0;I2=0;I1=1;I0=0; #20;
                I3=0;I2=0;I1=1;I0=1; #20;
                I3=0;I2=1;I1=0;I0=0; #20;
                I3=0;I2=1;I1=0;I0=1; #20;
                I3=0;I2=1;I1=1;I0=0; #20;
                I3=0;I2=1;I1=1;I0=1; #20;

                S1=0;S0=1;
                I3=1;I2=0;I1=0;I0=0; #20;
                I3=1;I2=0;I1=0;I0=1; #20;
                I3=1;I2=0;I1=1;I0=0; #20;
                I3=1;I2=0;I1=1;I0=1; #20;
                I3=1;I2=1;I1=0;I0=0; #20;
                I3=1;I2=1;I1=0;I0=1; #20;
                I3=1;I2=1;I1=1;I0=0; #20;
                I3=1;I2=1;I1=1;I0=1; #20;
```



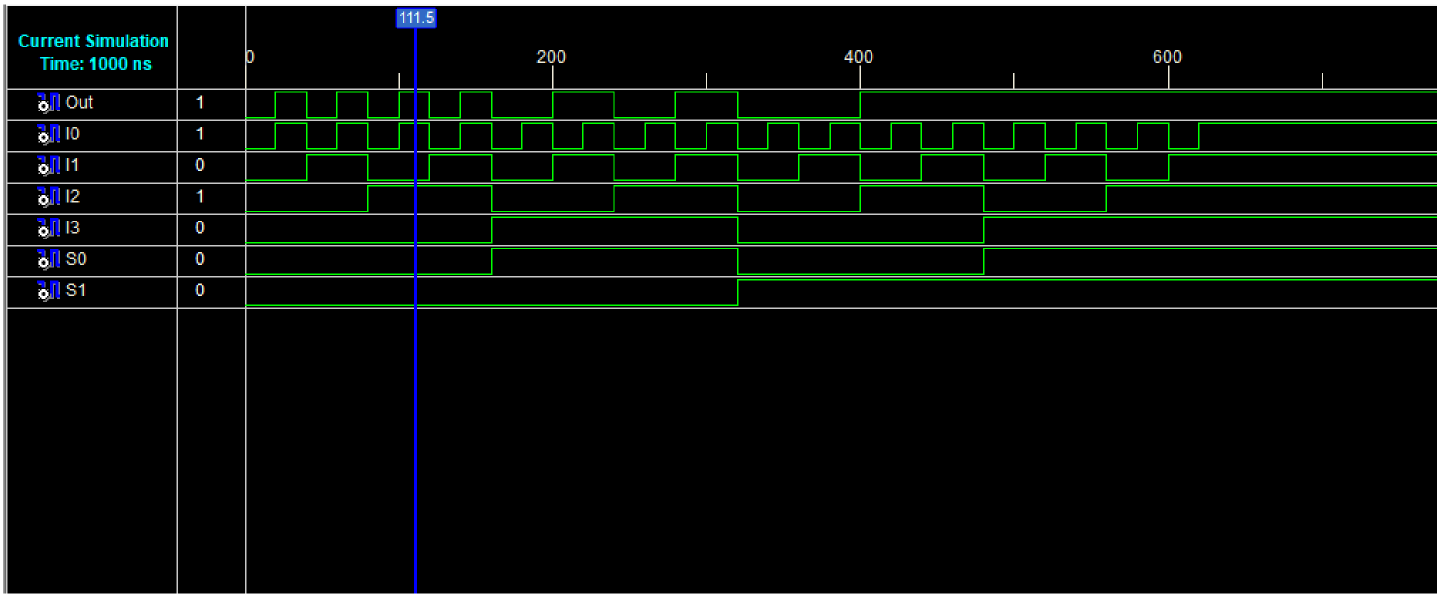
```
                I3=1;I2=0;I1=1;I0=1;  #20;
                I3=1;I2=1;I1=0;I0=0;  #20;
                I3=1;I2=1;I1=0;I0=1;  #20;
                I3=1;I2=1;I1=1;I0=0;  #20;
                I3=1;I2=1;I1=1;I0=1;  #20;

                S1=1;S0=0;
                I3=0;I2=0;I1=0;I0=0;  #20;
                I3=0;I2=0;I1=0;I0=1;  #20;
                I3=0;I2=0;I1=1;I0=0;  #20;
                I3=0;I2=0;I1=1;I0=1;  #20;
                I3=0;I2=1;I1=0;I0=0;  #20;
                I3=0;I2=1;I1=0;I0=1;  #20;
                I3=0;I2=1;I1=1;I0=0;  #20;
                I3=0;I2=1;I1=1;I0=1;  #20;

                S1=1;S0=1;
                I3=1;I2=0;I1=0;I0=0;  #20;
                I3=1;I2=0;I1=0;I0=1;  #20;
                I3=1;I2=0;I1=1;I0=0;  #20;
                I3=1;I2=0;I1=1;I0=1;  #20;
                I3=1;I2=1;I1=0;I0=0;  #20;
                I3=1;I2=1;I1=0;I0=1;  #20;
                I3=1;I2=1;I1=1;I0=0;  #20;
                I3=1;I2=1;I1=1;I0=1;  #20;

end

endmodule
```

Circuit





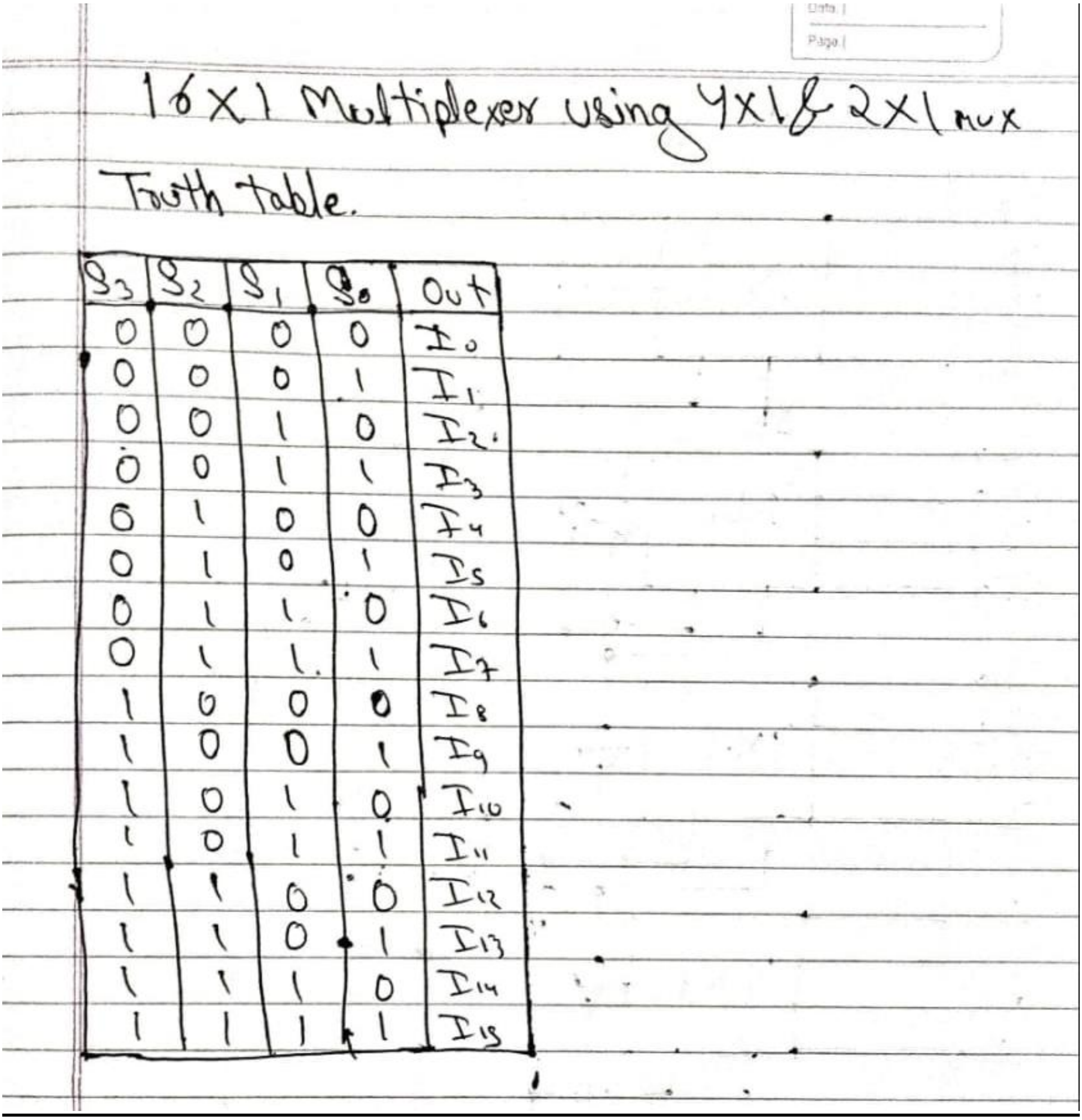
Waveform



Current Simulation Time: 1000 ns

Out	1
I0	1
I1	0
I2	1
I3	0
S0	0
S1	0

16x1 Multiplexer using 2x1 & 4x1 Multiplexer



Date |

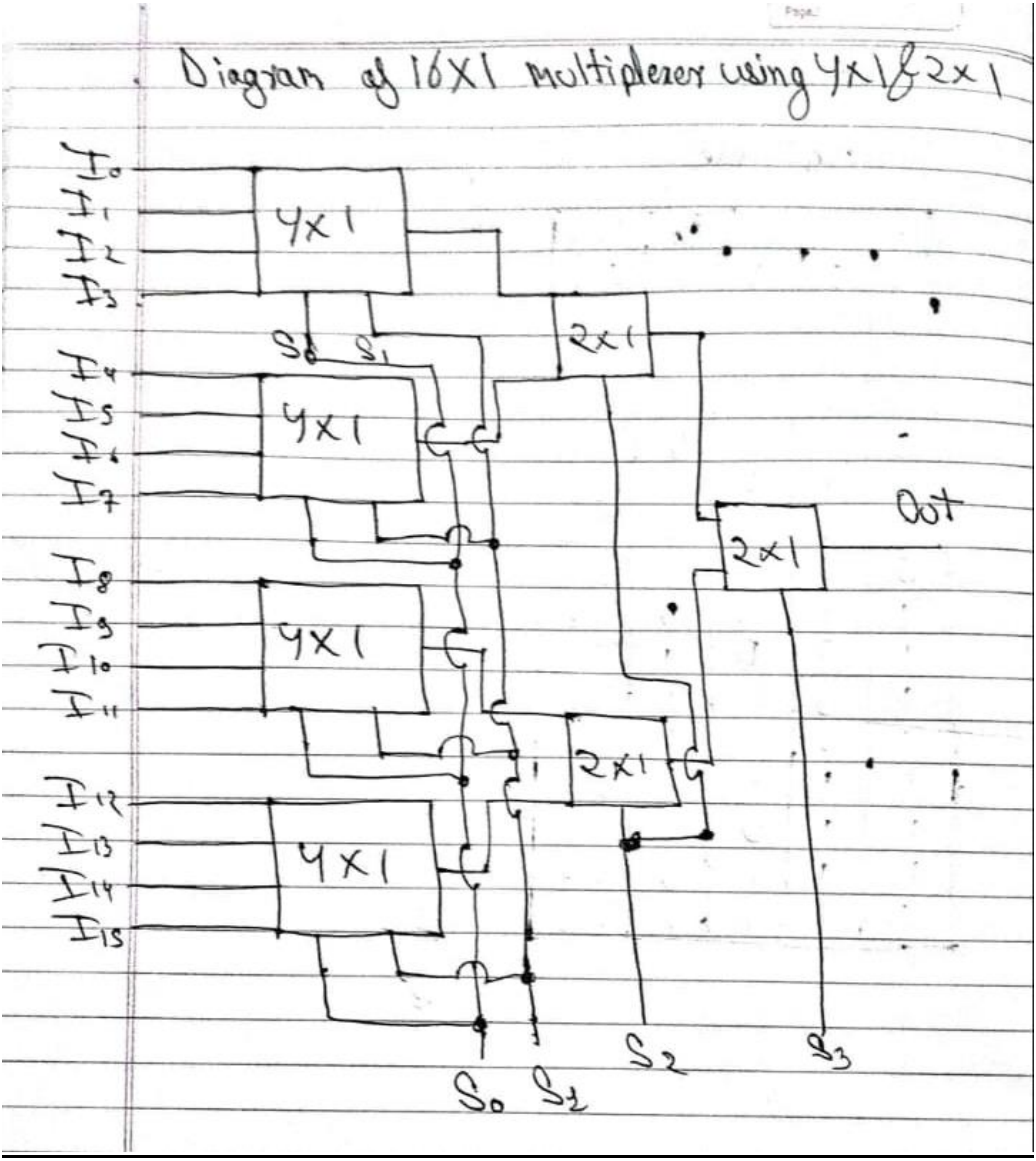
Page |

16x1 Multiplexer using 4x1 & 2x1 mux

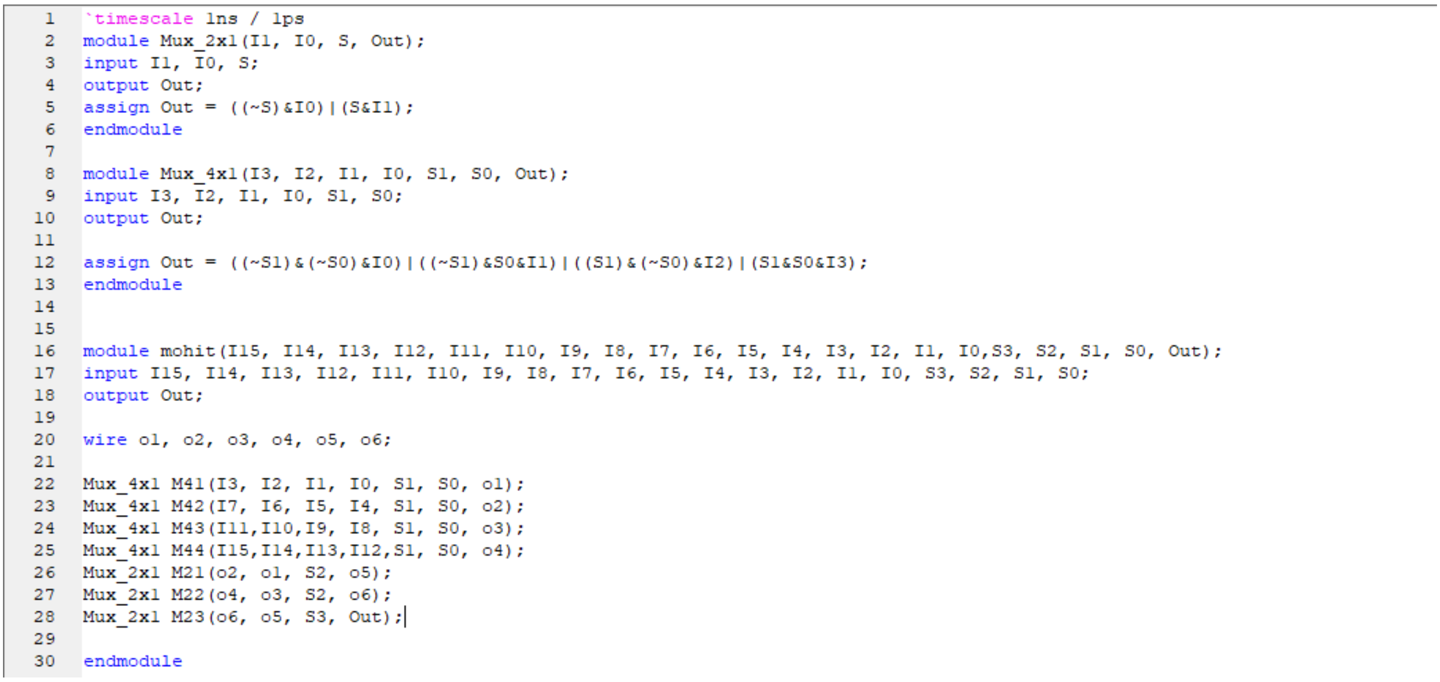
Truth table.

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}

Diagram of 16x1 using 2x1 & 4x1

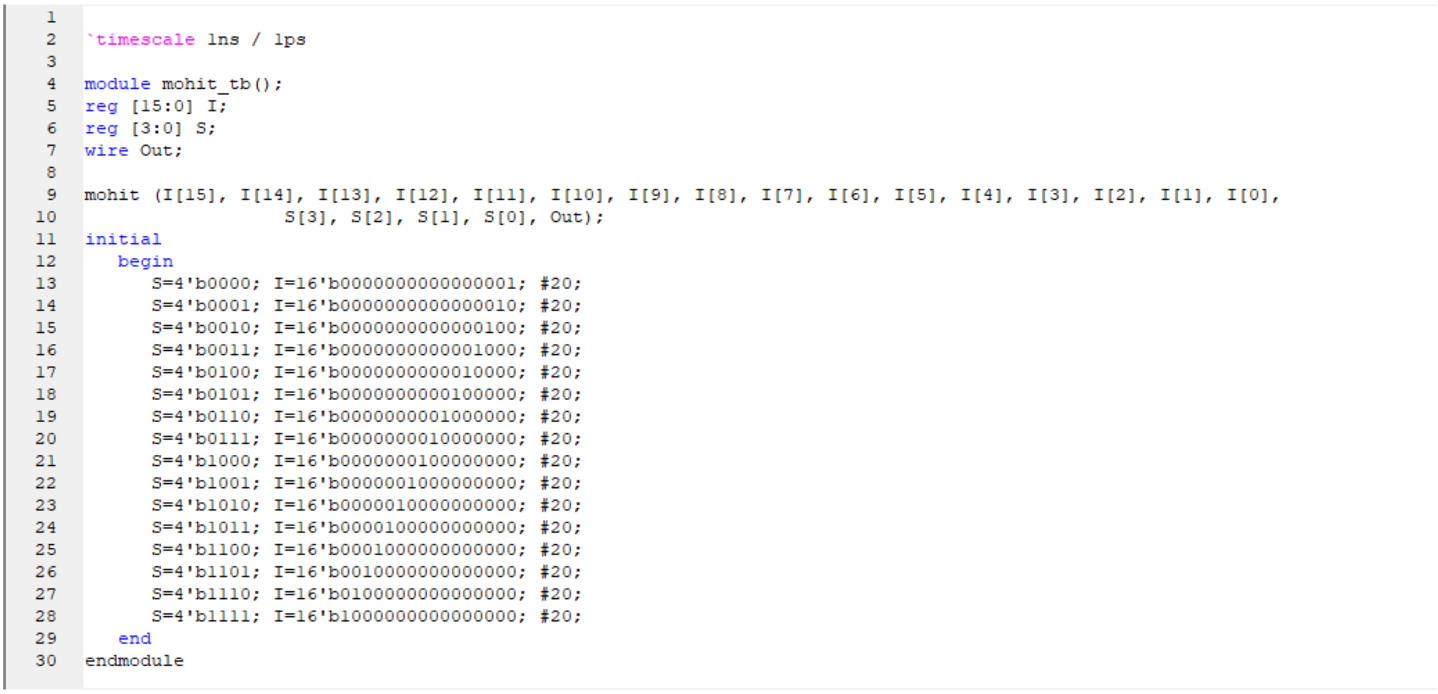


Verilog Code



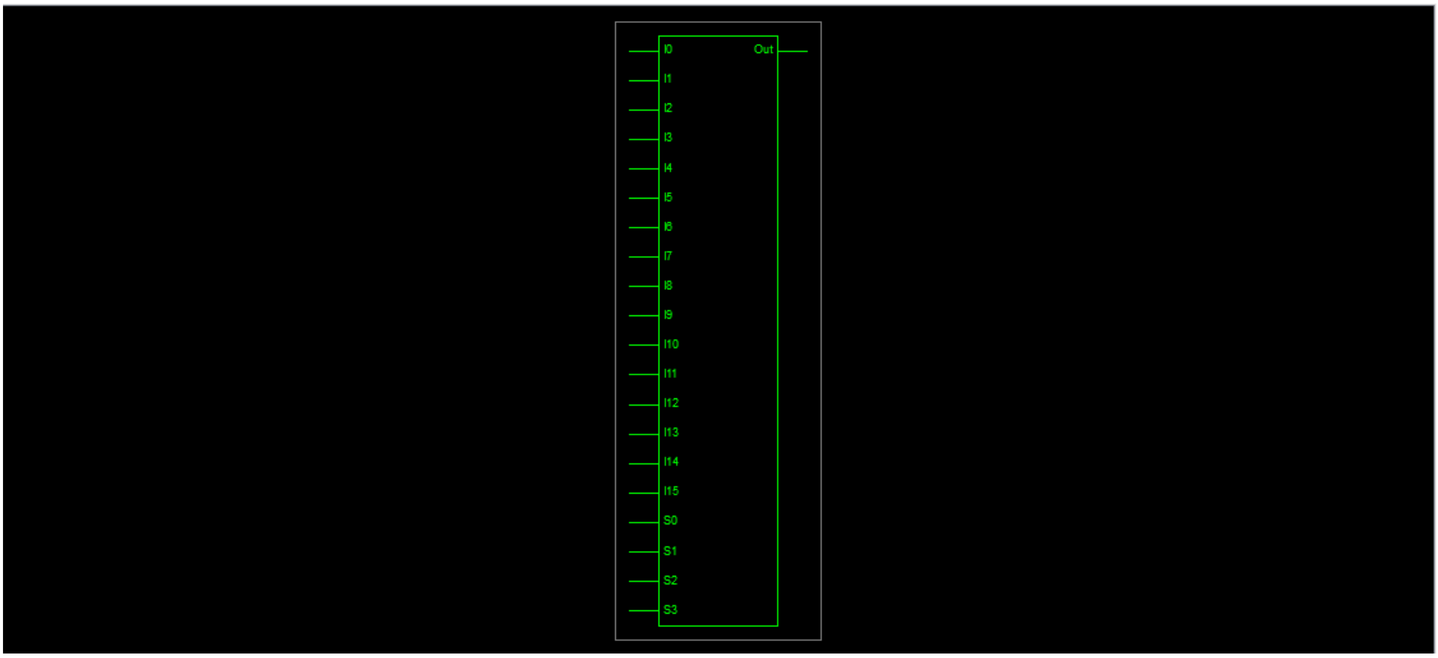
```verilog
`timescale 1ns / 1ps
module Mux_2x1(I1, I0, S, Out);
input I1, I0, S;
output Out;
assign Out = ((~S)&I0)|(S&I1);
endmodule

module Mux_4x1(I3, I2, I1, I0, S1, S0, Out);
input I3, I2, I1, I0, S1, S0;
output Out;

assign Out = ((~S1)&(~S0)&I0)|((~S1)&S0&I1)|((S1)&(~S0)&I2)|(S1&S0&I3);
endmodule


module mohit(I15, I14, I13, I12, I11, I10, I9, I8, I7, I6, I5, I4, I3, I2, I1, I0,S3, S2, S1, S0, Out);
input I15, I14, I13, I12, I11, I10, I9, I8, I7, I6, I5, I4, I3, I2, I1, I0, S3, S2, S1, S0;
output Out;

wire o1, o2, o3, o4, o5, o6;

Mux_4x1 M41(I3, I2, I1, I0, S1, S0, o1);
Mux_4x1 M42(I7, I6, I5, I4, S1, S0, o2);
Mux_4x1 M43(I11,I10,I9, I8, S1, S0, o3);
Mux_4x1 M44(I15,I14,I13,I12,S1, S0, o4);
Mux_2x1 M21(o2, o1, S2, o5);
Mux_2x1 M22(o4, o3, S2, o6);
Mux_2x1 M23(o6, o5, S3, Out);

endmodule
```

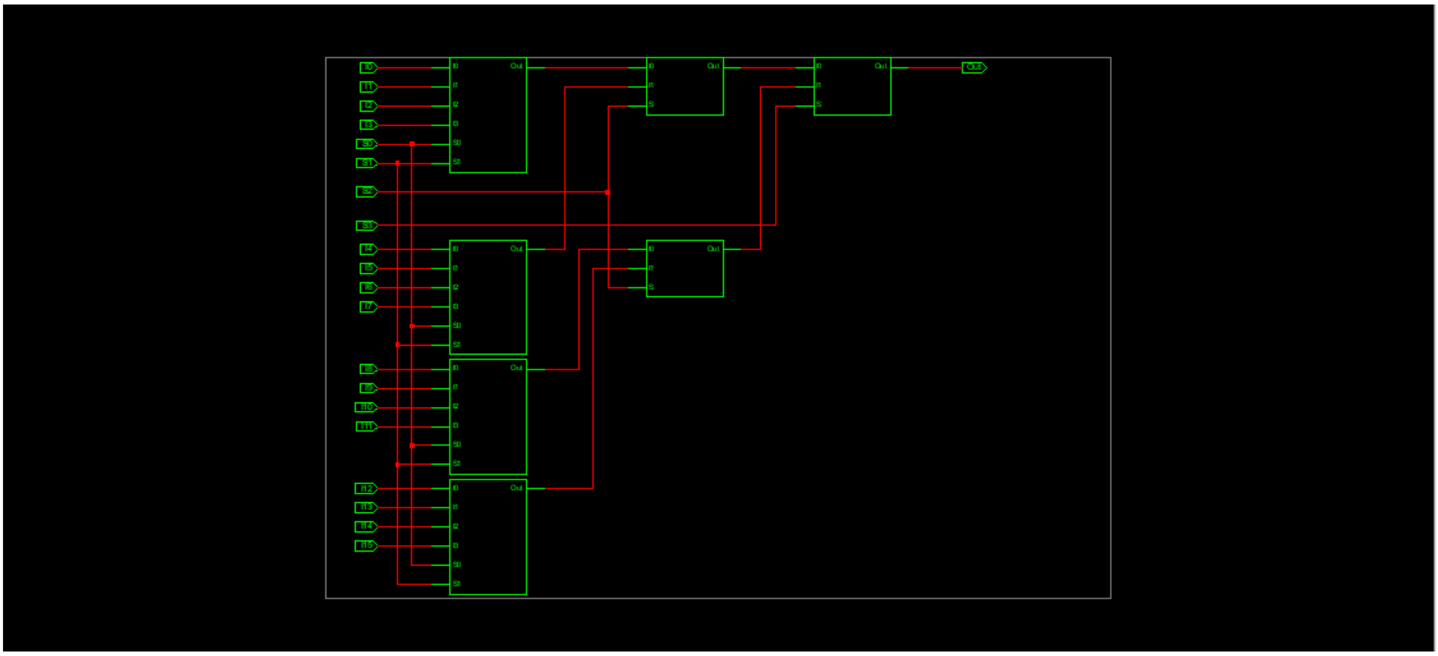
Test bench Program

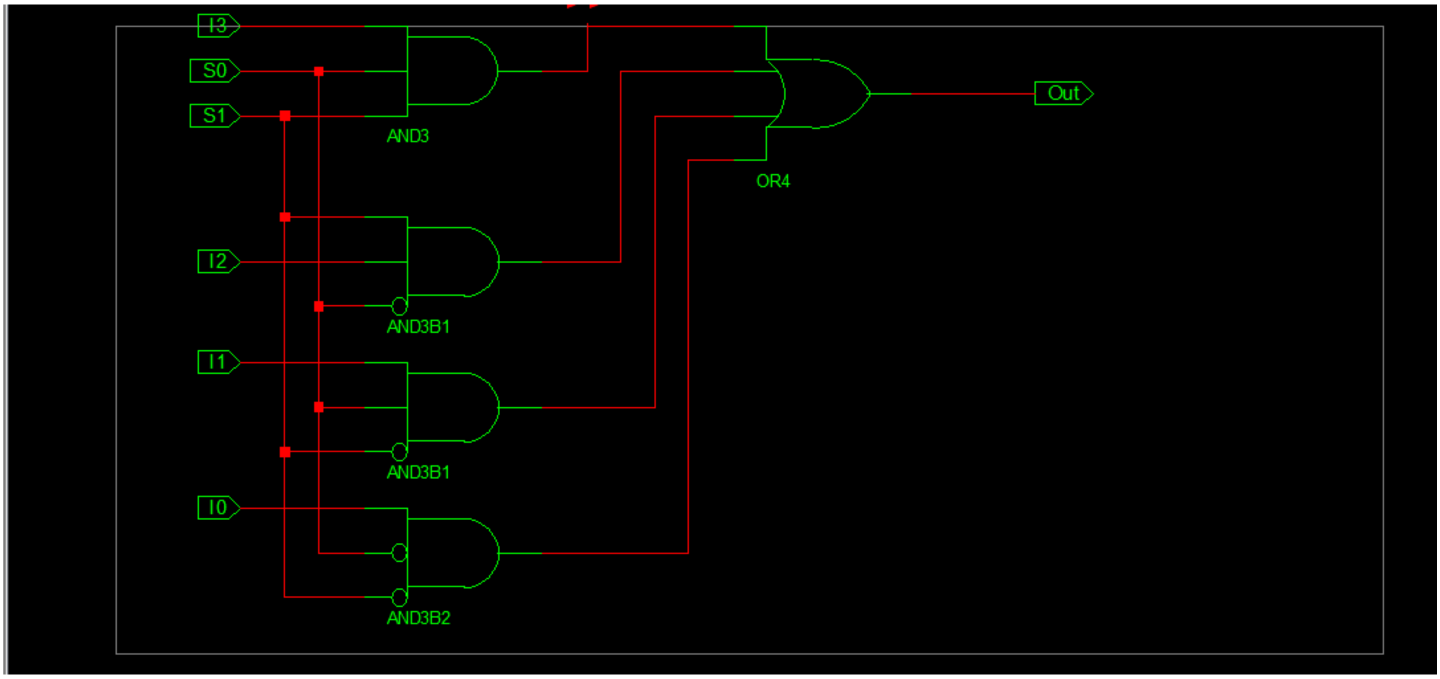


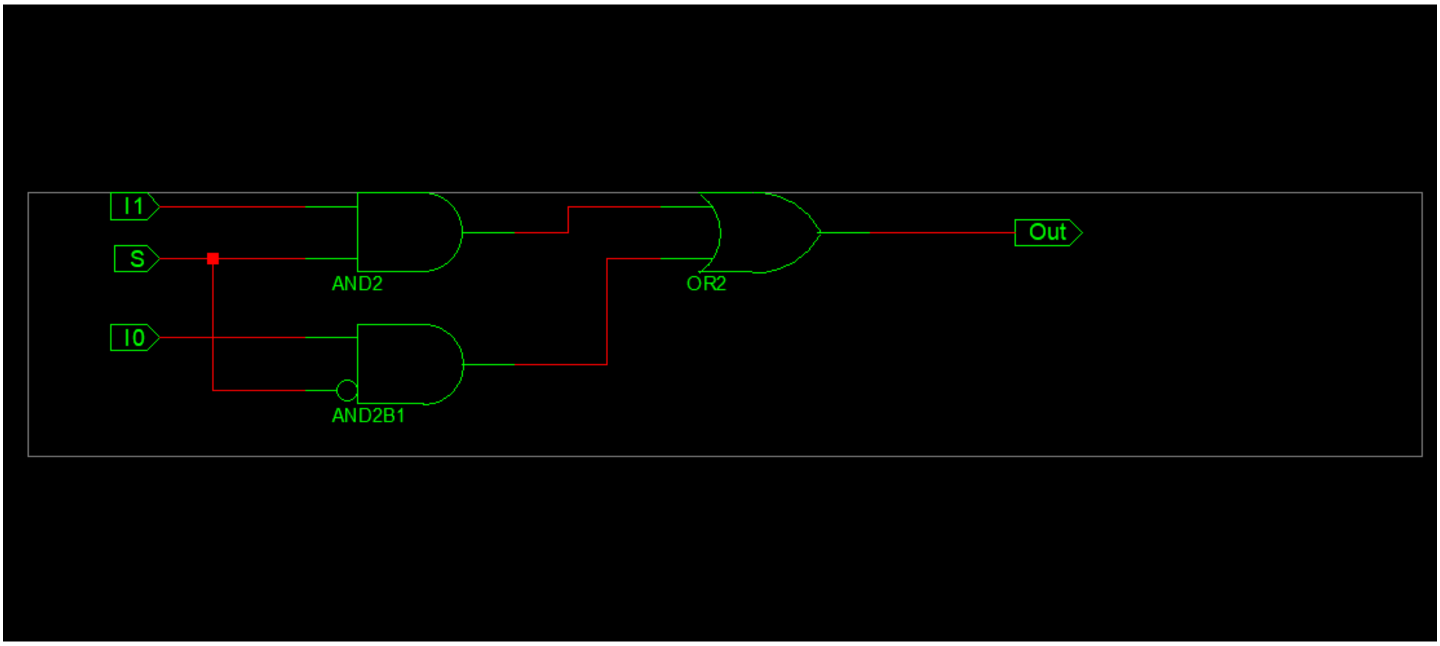
```verilog

`timescale 1ns / 1ps

module mohit_tb();
reg [15:0] I;
reg [3:0] S;
wire Out;

mohit  (I[15], I[14], I[13], I[12], I[11], I[10], I[9], I[8], I[7], I[6], I[5], I[4], I[3], I[2], I[1], I[0],
                  S[3], S[2], S[1], S[0], Out);
initial
   begin
      S=4'b0000; I=16'b0000000000000001; #20;
      S=4'b0001; I=16'b0000000000000010; #20;
      S=4'b0010; I=16'b0000000000000100; #20;
      S=4'b0011; I=16'b0000000000001000; #20;
      S=4'b0100; I=16'b0000000000010000; #20;
      S=4'b0101; I=16'b0000000000100000; #20;
      S=4'b0110; I=16'b0000000001000000; #20;
      S=4'b0111; I=16'b0000000010000000; #20;
      S=4'b1000; I=16'b0000000100000000; #20;
      S=4'b1001; I=16'b0000001000000000; #20;
      S=4'b1010; I=16'b0000010000000000; #20;
      S=4'b1011; I=16'b0000100000000000; #20;
      S=4'b1100; I=16'b0001000000000000; #20;
      S=4'b1101; I=16'b0010000000000000; #20;
      S=4'b1110; I=16'b0100000000000000; #20;
      S=4'b1111; I=16'b1000000000000000; #20;
   end
endmodule
```

Circuit Diagram

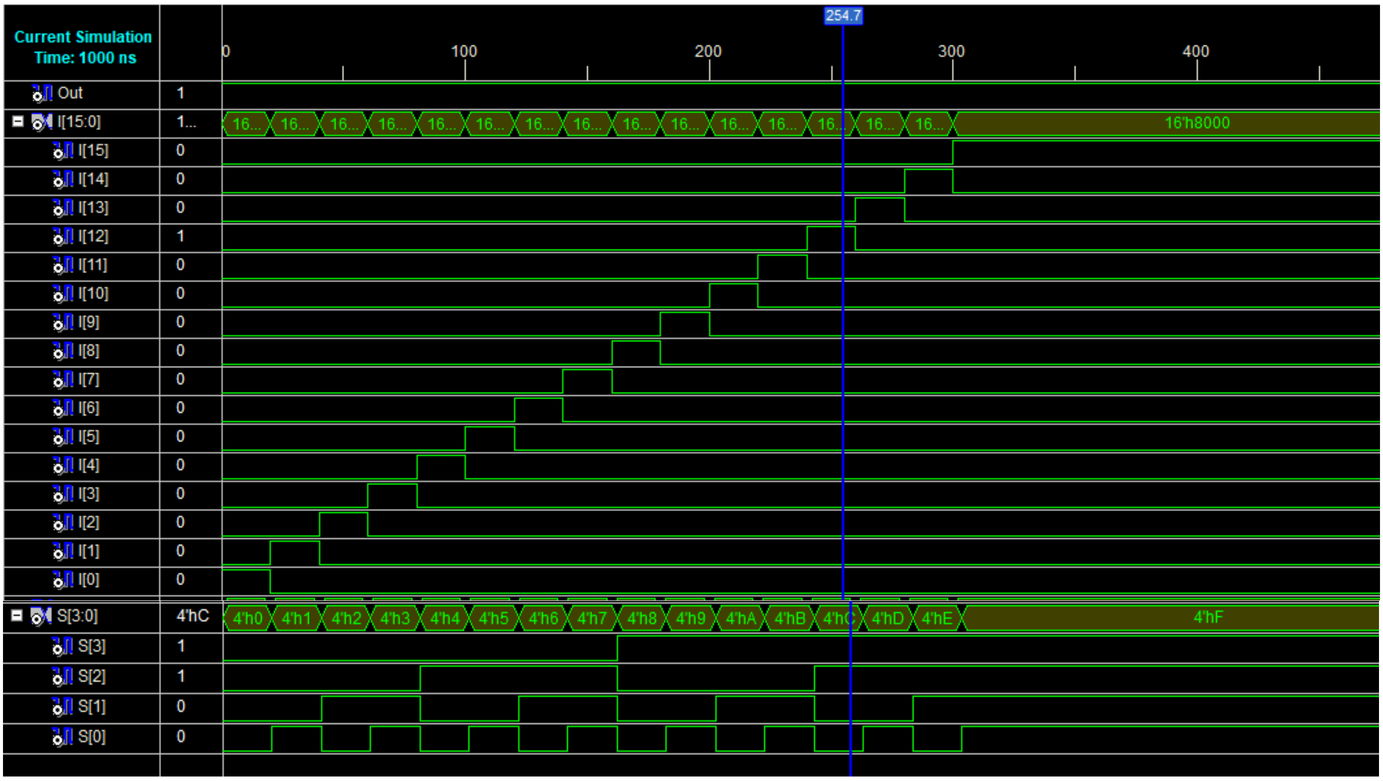





I3
S0
S1
AND3
Out
OR4
I2
AND3B1
I1
AND3B1
I0
AND3B2


I1
S
AND2
OR2
Out
I0
AND2B1

Waveform



END