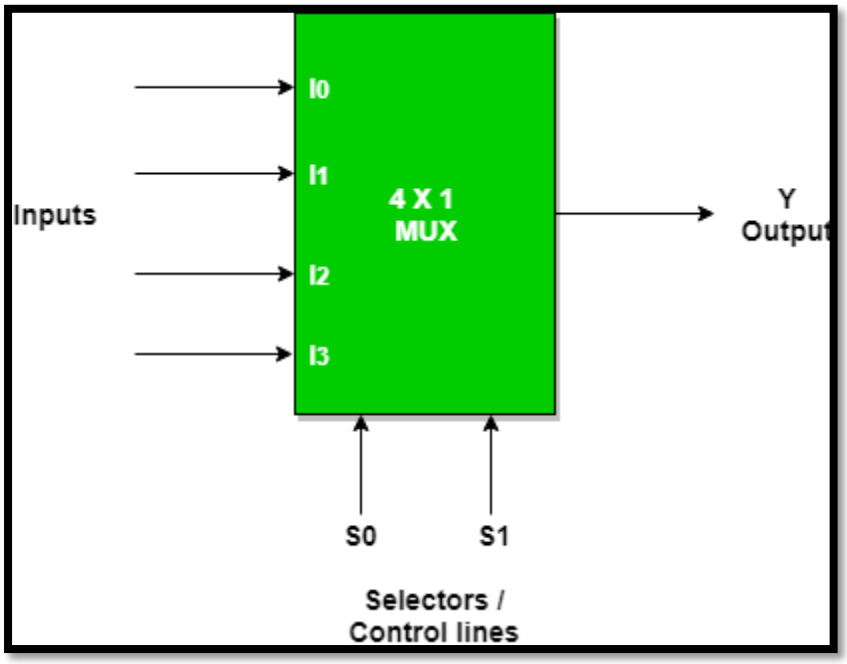
Multiplexers:

A multiplexer is a combinational circuit that has many data inputs and a single output, depending on control or select inputs. For N input lines, log2(N) selection lines are required, or equivalently, for 2^n input lines, n selection lines are needed. Multiplexers are also known as “N-to-1 selectors,” parallel-to-serial converters, many-to-one circuits, and universal logic circuits. They are mainly used to increase the amount of data that can be sent over a network within a certain amount of time and bandwidth.

4_x_1 Mux:

In this article we write Verilog HDL code for 4 to 1 mux as we know that the 4 to 1 mux is four inputs and one output, and the output will select as the basis of select lines.

Block Diagram:



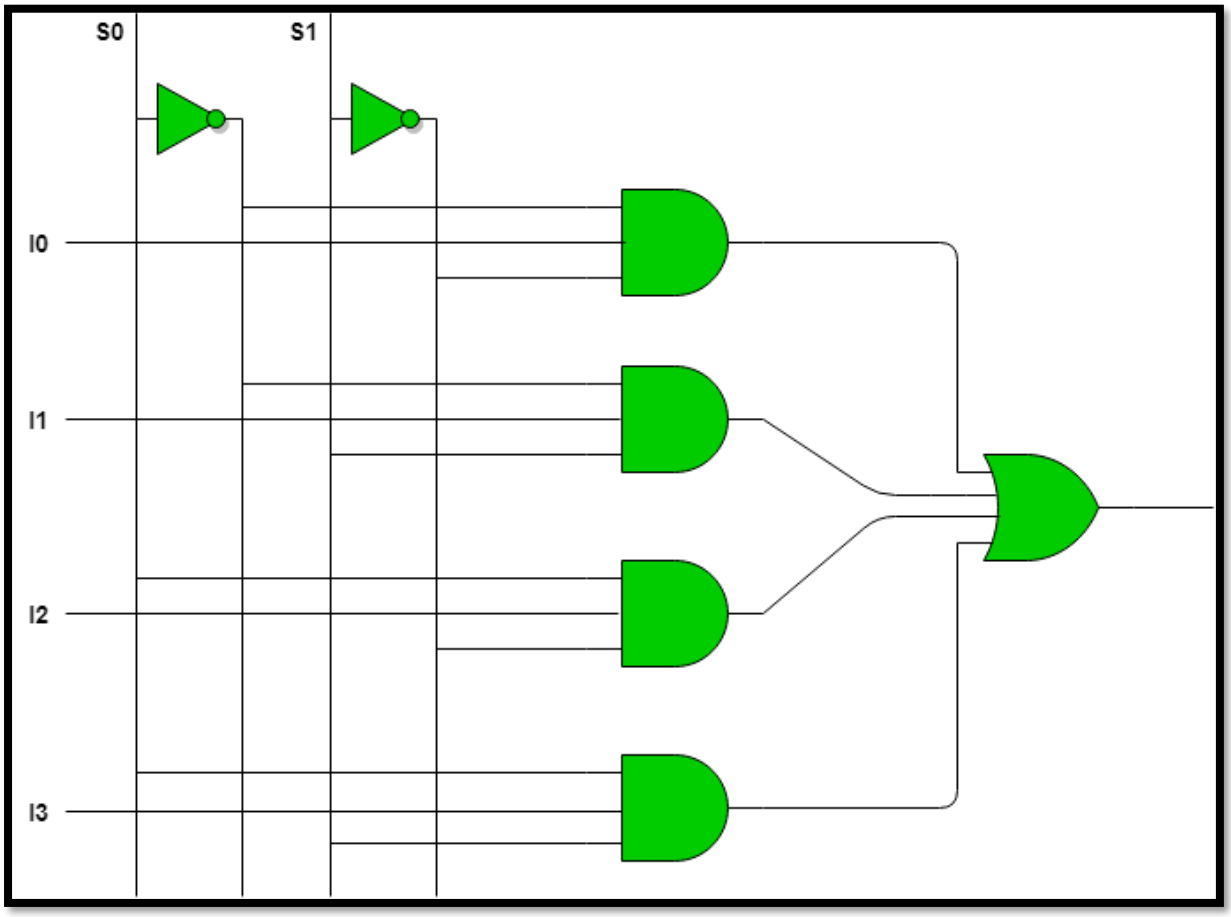
Truth Table:

Truth Table

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

So, final equation,
Y = S0'.S1'.I0 + S0'.S1.I1 + S0.S1'.I2 + S0.S1.I3

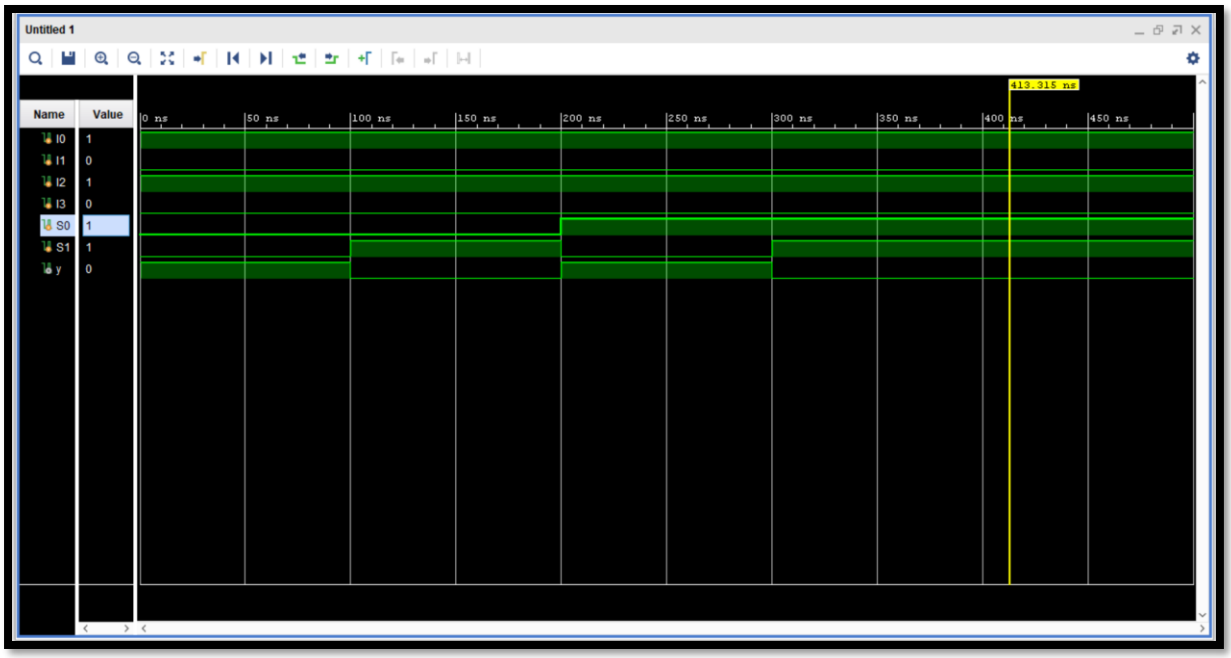
Logic Circuit:



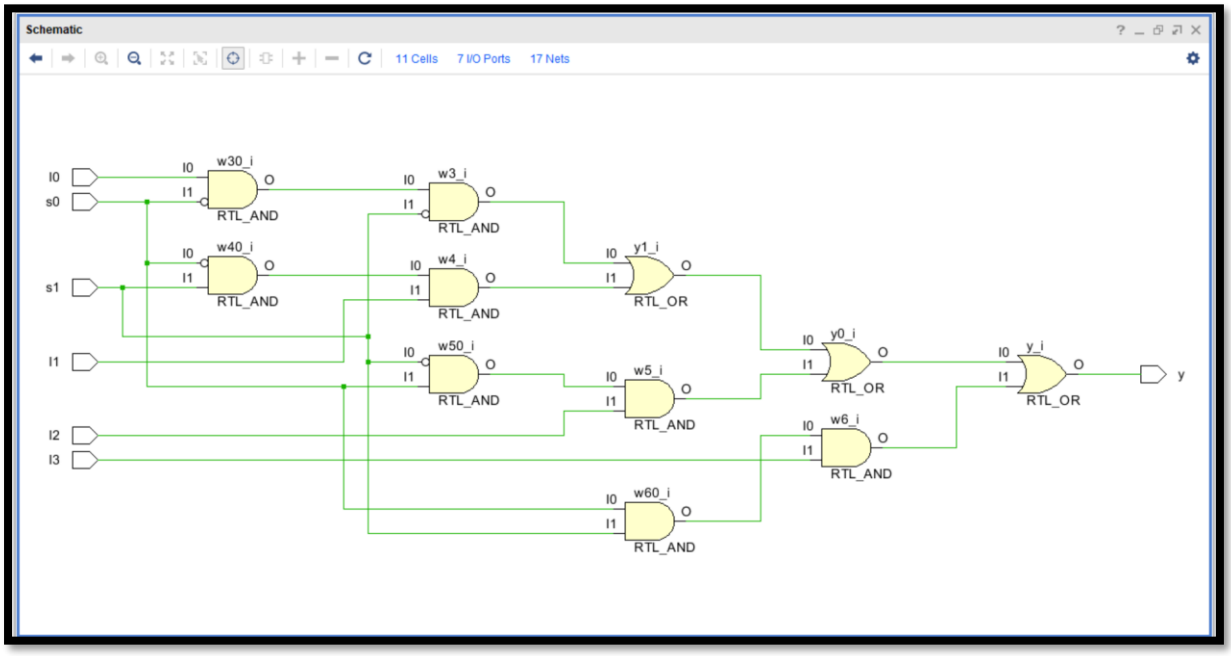
Verilog Code:	**Testbench:**

module mux_4_1(I0,I1,I2,I3,s0,s1,y);
input I0,I1,I2,I3,s0,s1;
output y;
wire w1,w2,w3,w4,w5,w6;
not(w1,s0);
not(w2,s1);
and(w3,I0,w1,w2);
and(w4,w1,s1,I1);
and(w5,w2,s0,I2);
and(w6,s0,s1,I3);
or(y,w3,w4,w5,w6);
endmodule
	module mux_4_1_tb();
	reg I0,I1,I2,I3,S0,S1;
	wire y;
	mux_4_1 m1(I0,I1,I2,I3,S0,S1,y);
	initial
	begin
	I0=1 ;I1=0 ;I2=1 ;I3=0;
	
	$display("I0=%b I1=%b I2=%b
	I3=%b\n",I0,I1,I2,I3);
	S0=0;S1=0;
	#100
	$display("Time=%d S0=%b S1=%b
	y=%b\n",$time,S0,S1,y);
	S0=0;S1=1;
	#100
	$display("Time=%d S0=%b S1=%b
	y=%b\n",$time,S0,S1,y);
	S0=1;S1=0;
	#100
	$display("Time=%d S0=%b S1=%b
	y=%b\n",$time,S0,S1,y);
	S0=1;S1=1;
	#100
	$display("Time=%d S0=%b S1=%b
	y=%b\n",$time,S0,S1,y);
	#100 $stop;
	end
	endmodule

Simulation:



RTL design



Tcl Console Output: