SIMULATION AND IMPLEMENTATION OF COMBINATIONAL LOGIC CIRCUITS

AIM: To simulate and synthesis ENCODER, DECODER, MULTIPLEXER, DEMULTIPLEXER, MAGNITUDE COMPARATOR using Xilinx ISE.

APPARATUS REQUIRED: Xilinx 14.7 Spartan6 FPGA

PROCEDURE:

STEP:1 Start the Xilinx navigator, Select and Name the New project.

STEP:2 Select the device family, device, package and speed.
STEP:3 Select new source in the New Project and select Verilog Module as the Source type.
STEP:4 Type the File Name and Click Next and then finish button. Type the code and save it. STEP:5 Select the Behavioral Simulation in the Source Window and click the check syntax.
STEP:6 Click the simulation to simulate the program and give the inputs and verify the outputs as per the truth table.
STEP:7 Select the Implementation in the Sources Window and select the required file in the Processes Window.

STEP:8 Select Check Syntax from the Synthesize XST Process. Double Click in the FloorplanArea/IO/Logic-Post Synthesis process in the User Constraints process group. UCF(User constraint File) is obtained.

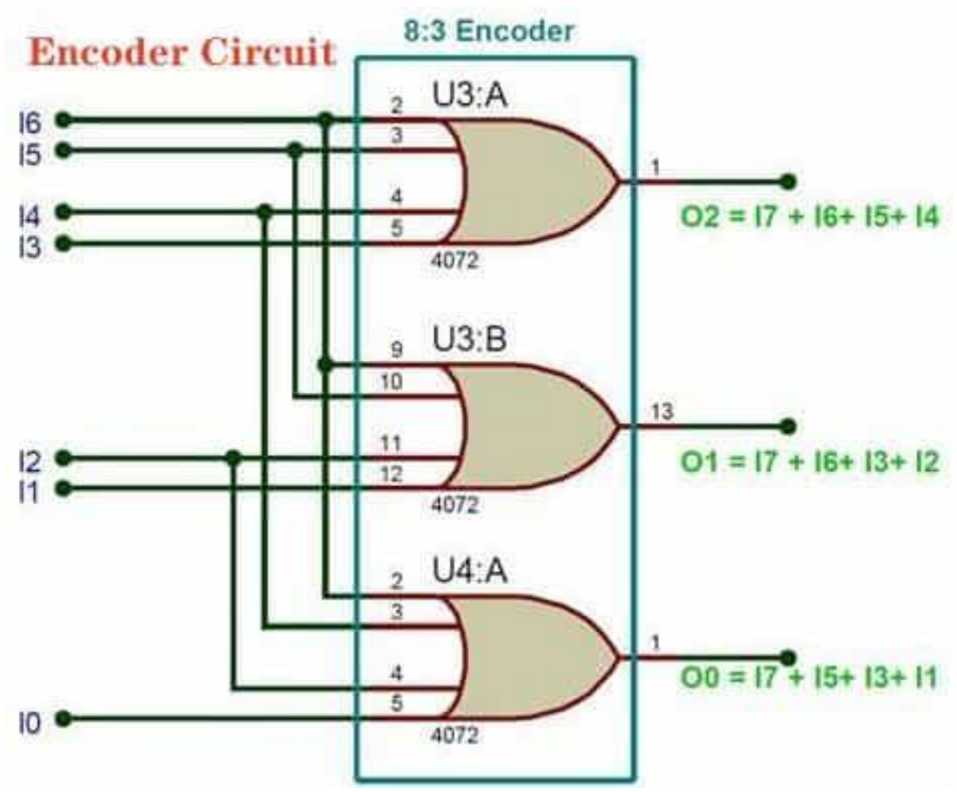
STEP:9 In the Design Object List Window, enter the pin location for each pin in the Loc column Select save from the File menu.

STEP:10 Double click on the Implement Design and double click on the Generate Programming File to create a bitstream of the design.(.v) file is converted into .bit file here.

STEP:11 On the board, by giving required input, the LEDs starts to glow light, indicating the output.

LOGIC DIAGRAM

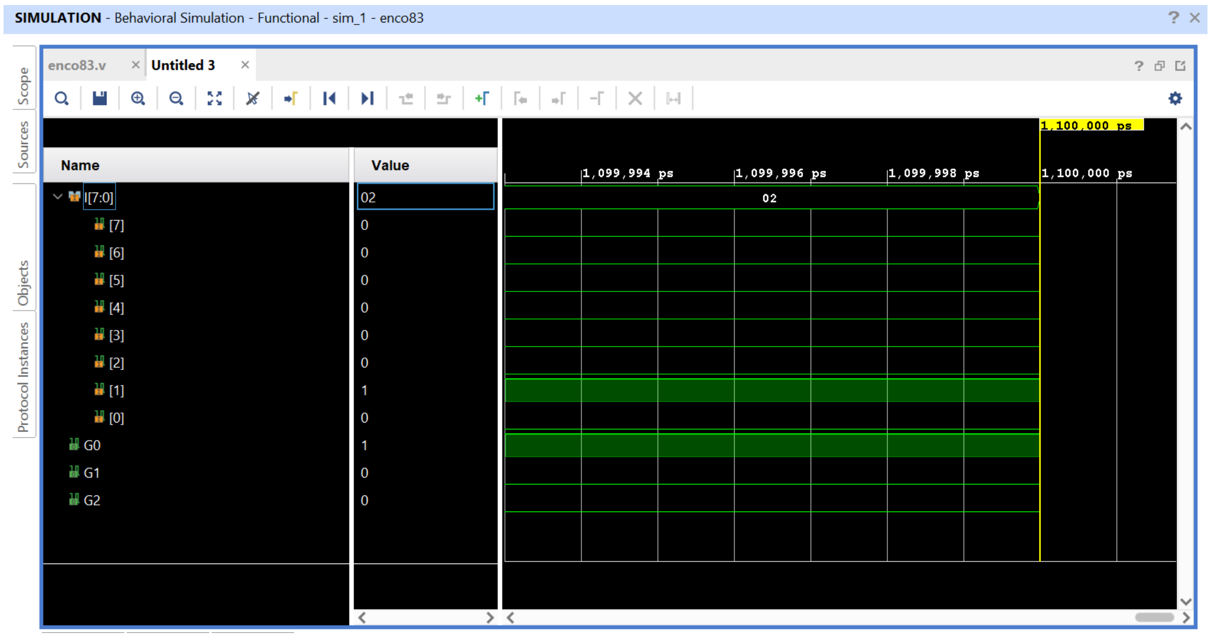
ENCODER



VERILOG CODE

```
module enco83(I,G0,G1,G2);
input [7:0]I;
output G0,G1,G2;
or g1(G0,I[1],I[3],I[5],I[7]);
or g2(G1,I[2],I[3],I[6],I[7]);
or g3(G2,I[7],I[6],I[5],I[4]);
endmodule
```

OUTPUT WAVEFORM

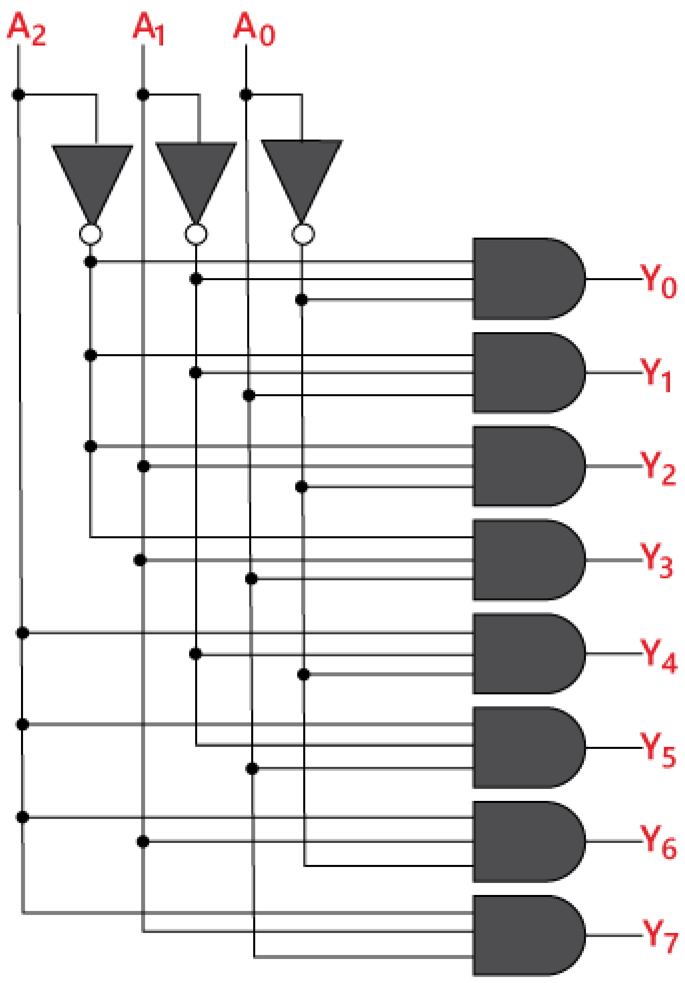


SIMULATION - Behavioral Simulation - Functional - sim_1 - enco83

enco83.v | Untitled 3

Name	Value
I[7:0]	02
[7]	0
[6]	0
[5]	0
[4]	0
[3]	0
[2]	0
[1]	1
[0]	0
G0	1
G1	0
G2	0

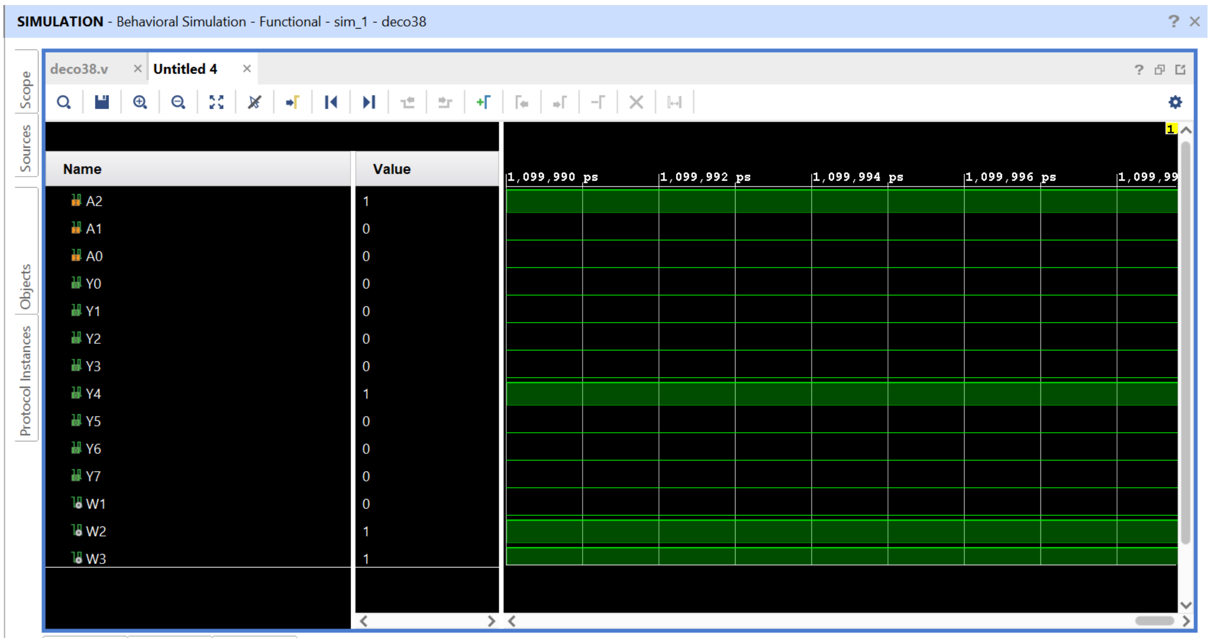
DECODER



VERILOG CODE

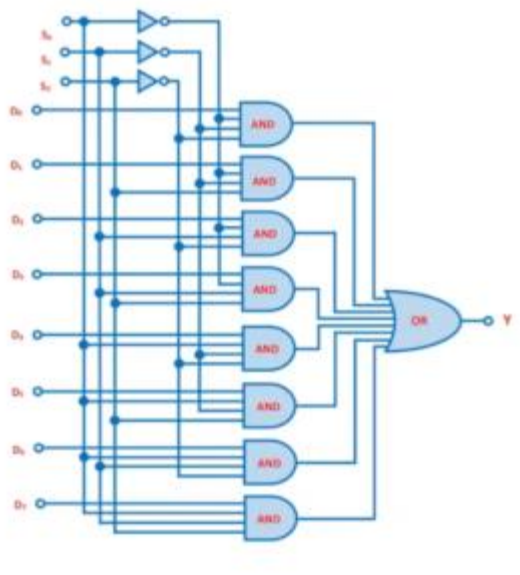
```
module deco38(A2,A1,A0,Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7);
input A2,A1,A0;
output Y0,Y1,Y2,Y3,Y4,Y5,Y6,Y7;
wire W1,W2,W3;
not g1(W1,A2);
not g2(W2,A1);
not g3(W3,A0);
and g4(Y0,W1,W2,W3);
and g5(Y1,W1,W2,A0);
and g6(Y2,W1,A1,W3);
and g7(Y3,W1,A1,A0);
and g8(Y4,A2,W2,W3);
and g9(Y5,A2,W2,A0);
and g10(Y6,A2,A1,W3);
and g11(Y7,A2,A1,A0);
endmodule
```

OUTPUT WAVEFORM



Name	Value
A2	1
A1	0
A0	0
Y0	0
Y1	0
Y2	0
Y3	0
Y4	1
Y5	0
Y6	0
Y7	0
W1	0
W2	1
W3	1

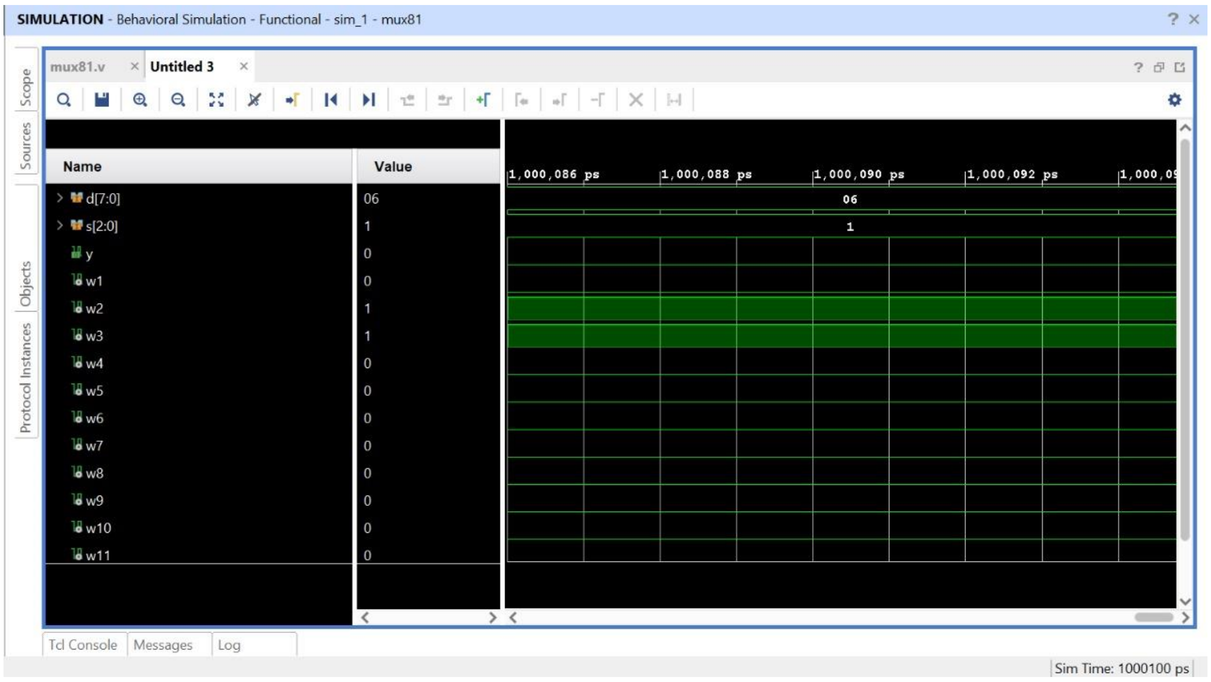
MULTIPLEXER



VERILOG CODE

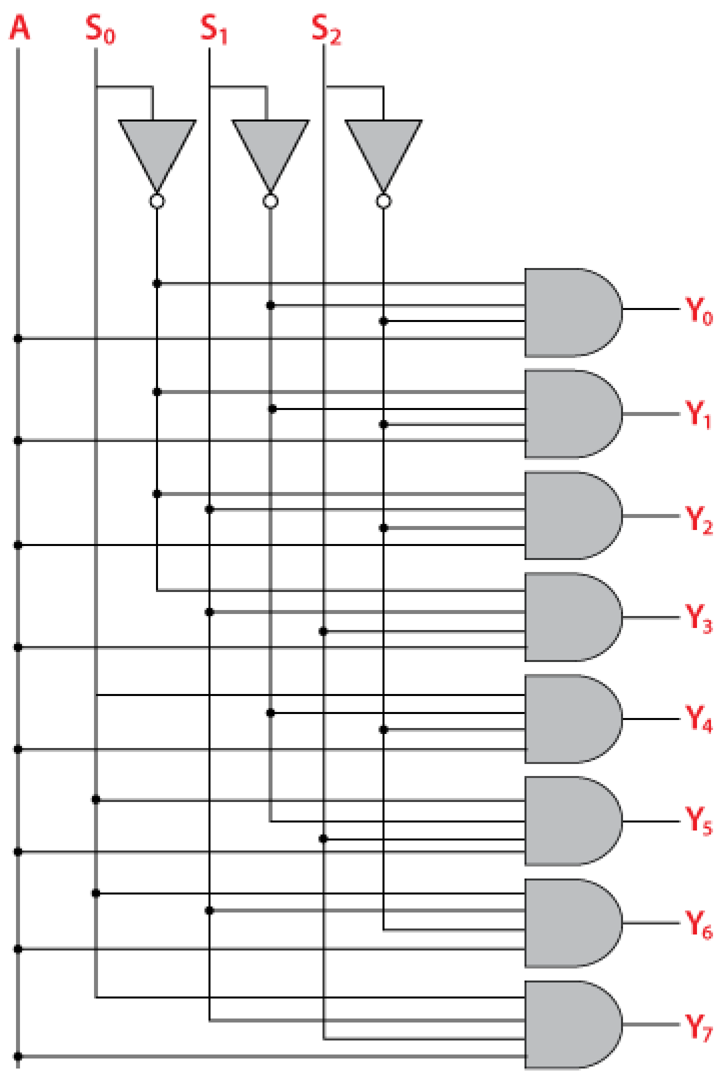
```
module mux81(d,s,y);
input [7:0]d;
input [2:0]s;
output y;
wire w1,w2,w3,w4,w5,w6,w7,w8,w9,w10,w11;
not g1(w1,s[0]);
not g2(w2,s[1]);
not g3(w3,s[2]);
and g4(w4,d[0],w1,w2,w3);
and g5(w5,d[1],w1,w2,s[2]);
and g6(w6,d[2],w1,w3,s[1]);
and g7(w7,d[3],w1,s[1],s[2]);
and g8(w8,d[4],s[0],w2,w3);
and g9(w9,d[5],s[0],s[2],w2);
and g10(w10,d[6],s[0],s[1],w3);
and g11(w11,d[7],s[0],s[1],s[2]);
or g12(y,w4,w5,w6,w8,w9,w10,w11);
endmodule
```

OUTPUT WAVEFORM


SIMULATION - Behavioral Simulation - Functional - sim_1 - mux81

Name	Value
d[7:0]	06
s[2:0]	1
y	0
w1	0
w2	1
w3	1
w4	0
w5	0
w6	0
w7	0
w8	0
w9	0
w10	0
w11	0

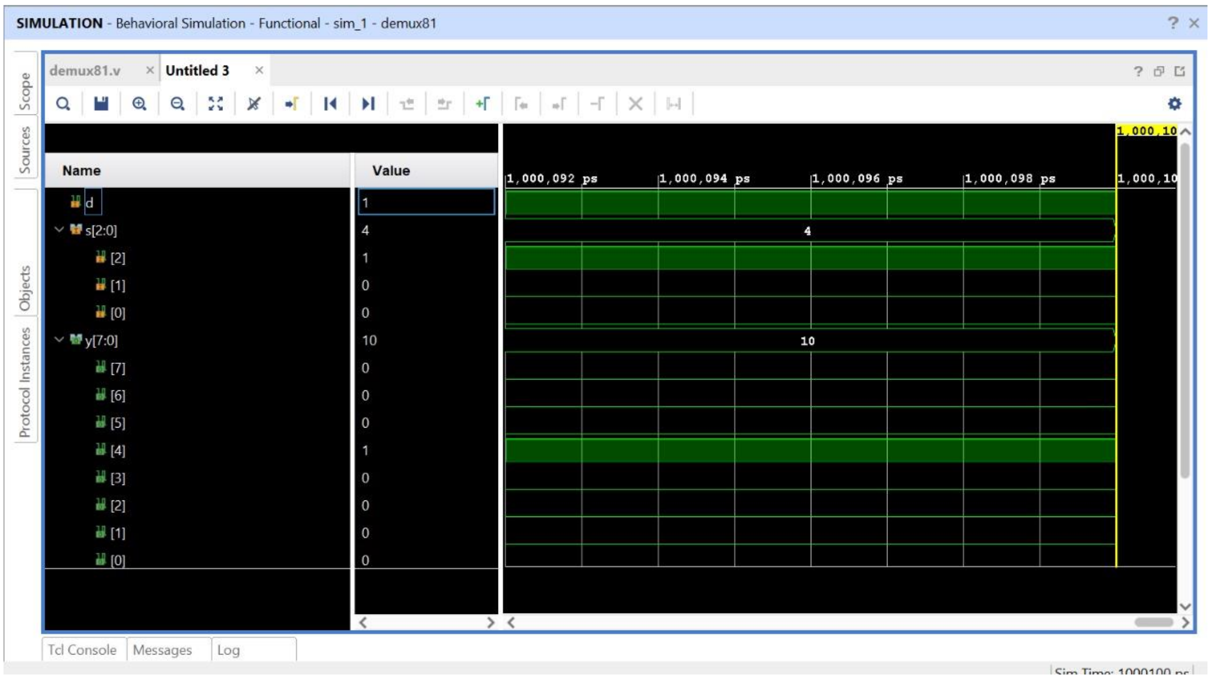
DEMULTIPLEXER



VERILOG CODE

```
module demux81(d,s,y);
input d;
input [2:0]s;
output [7:0]y;
wire w1,w2,w3;
not g1(w1,s[0]);
not g2(w2,s[1]);
not g3(w3,s[2]);
and g4(y[0],d,w1,w2,w3);
and g5(y[1],d,w1,s[0],w3);
and g6(y[2],d,w3,s[1],w1);
and g7(y[3],d,s[0],s[1],w3);
and g8(y[4],d,s[2],w1,w2);
and g9(y[5],d,s[2],s[0],w2);
and g10(y[6],d,w1,s[1],s[2]);
and g11(y[7],d,s[2],s[1],s[0]);
endmodule
```

OUTPUT WAVEFORM

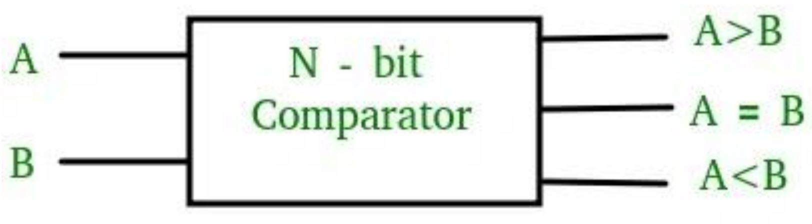


Name	Value
d	1
s[2:0]	4
[2]	1
[1]	0
[0]	0
y[7:0]	10
[7]	0
[6]	0
[5]	0
[4]	1
[3]	0
[2]	0
[1]	0
[0]	0

```
1,000,092 ps    1,000,094 ps    1,000,096 ps    1,000,098 ps    1,000,10
```

Sim Time: 1000100 ps

MAGNITUDE COMPARATOR

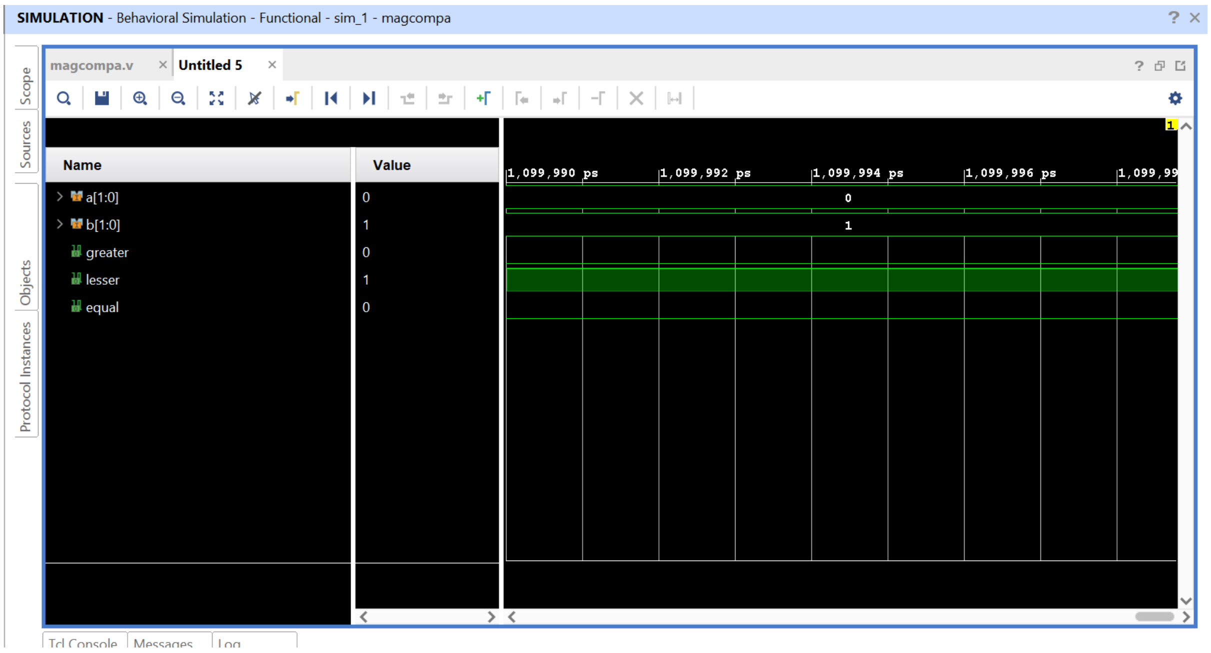


VERILOG CODE

```
module magcompa(a,b,greater,lesser,equal);
input [1:0]a,b;
output reg greater,lesser,equal;
always@(*)
begin
if(a<b)
begin
greater=1'b0;
equal=1'b0;
lesser=1'b1;
end
else if(a>b)
begin
```

```
greater=1'b1;
equal=1'b0;
lesser=1'b0;
end
else
begin
greater=1'b0;
equal=1'b1;
lesser=1'b0;
end
end
endmodule
```

OUTPUT WAVEFORM



Name	Value
a[1:0]	0
b[1:0]	1
greater	0
lesser	1
equal	0

RESULT

Thus the simulation and implementation of combinational logic circuits is verified successfully.