SIMULATION AND IMPLEMENTATION OF COMBINATIONAL LOGIC CIRCUITS

AIM:

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

APPARATUS REQUIRED:

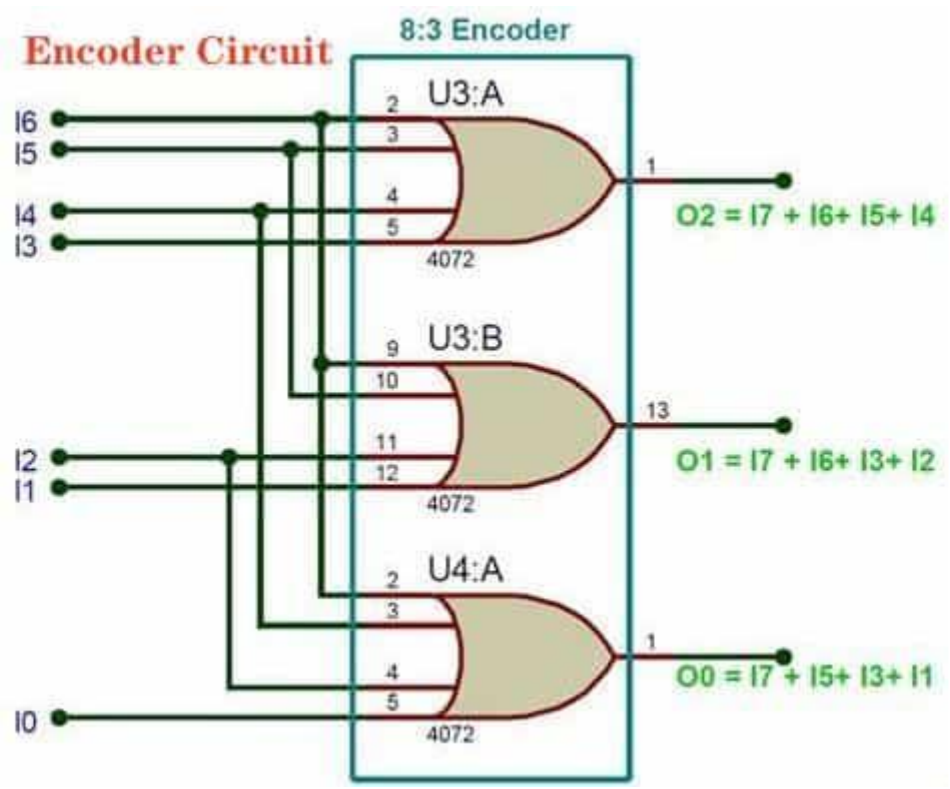
Xilinx 14.7
Spartan6 FPGA

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.

ENCODER

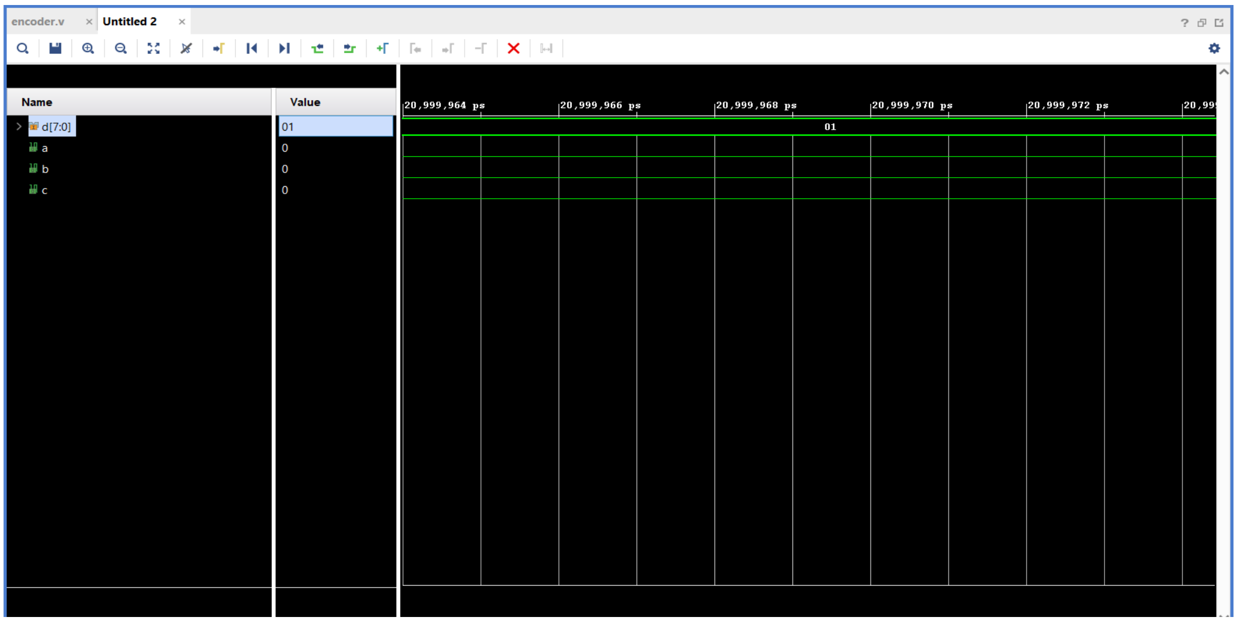
LOGIC DIAGRAM:



VERILOG CODE:

```
module encoder(d,y);
input[7:0]d;
output[2:0]y;
or g1(y[0],d[7],d[6],d[5],d[4]);
or g2(y[1],d[7],d[6],d[3],d[2]);
or g3(y[2],d[7],d[5],d[3],d[1]);
endmodule
```

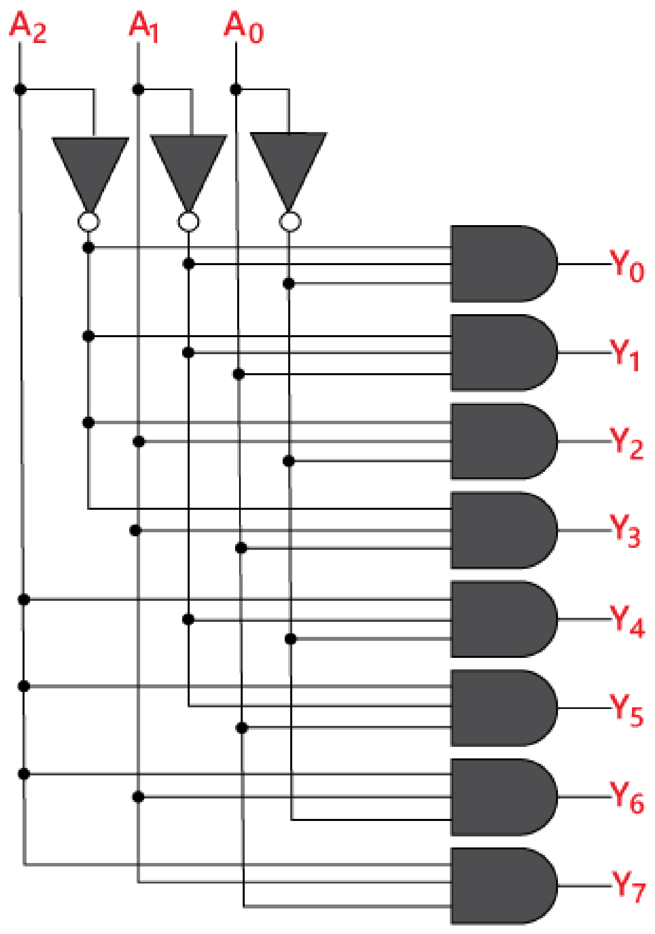
OUTPUT:



```
Name      Value
d[7:0]    01
a         0
b         0
c         0
```

DECODER

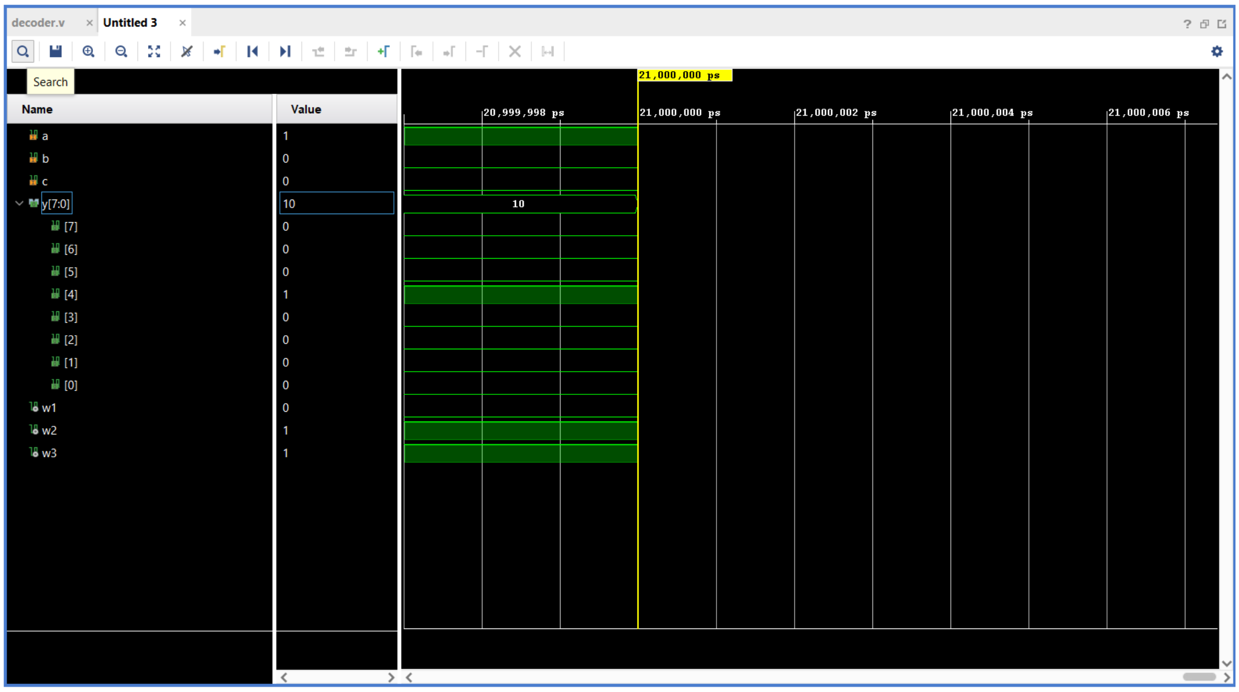
LOGIC DIAGRAM:



VERILOG CODE:

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

OUTPUT:



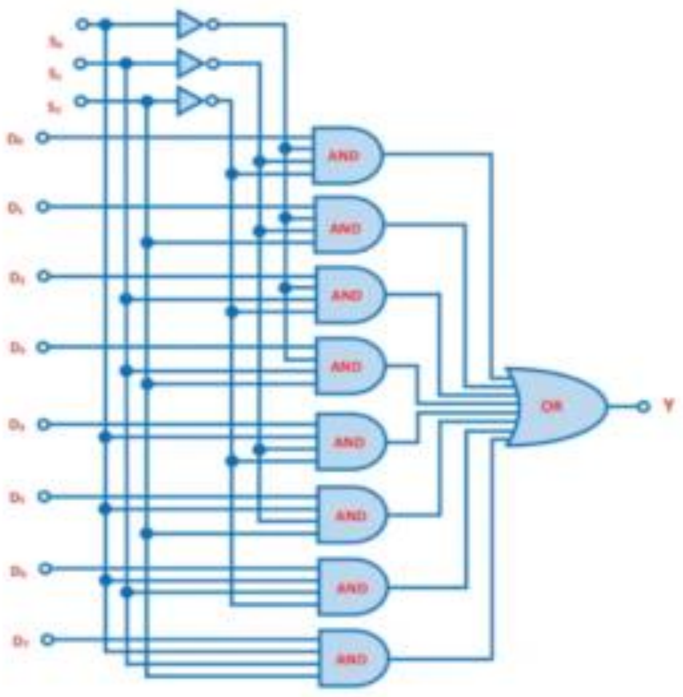
decoder.v × Untitled 3 ×

Search

Name	Value
a	1
b	0
c	0
y[7:0]	10
[7]	0
[6]	0
[5]	0
[4]	1
[3]	0
[2]	0
[1]	0
[0]	0
w1	0
w2	1
w3	1

MULTIPLEXER

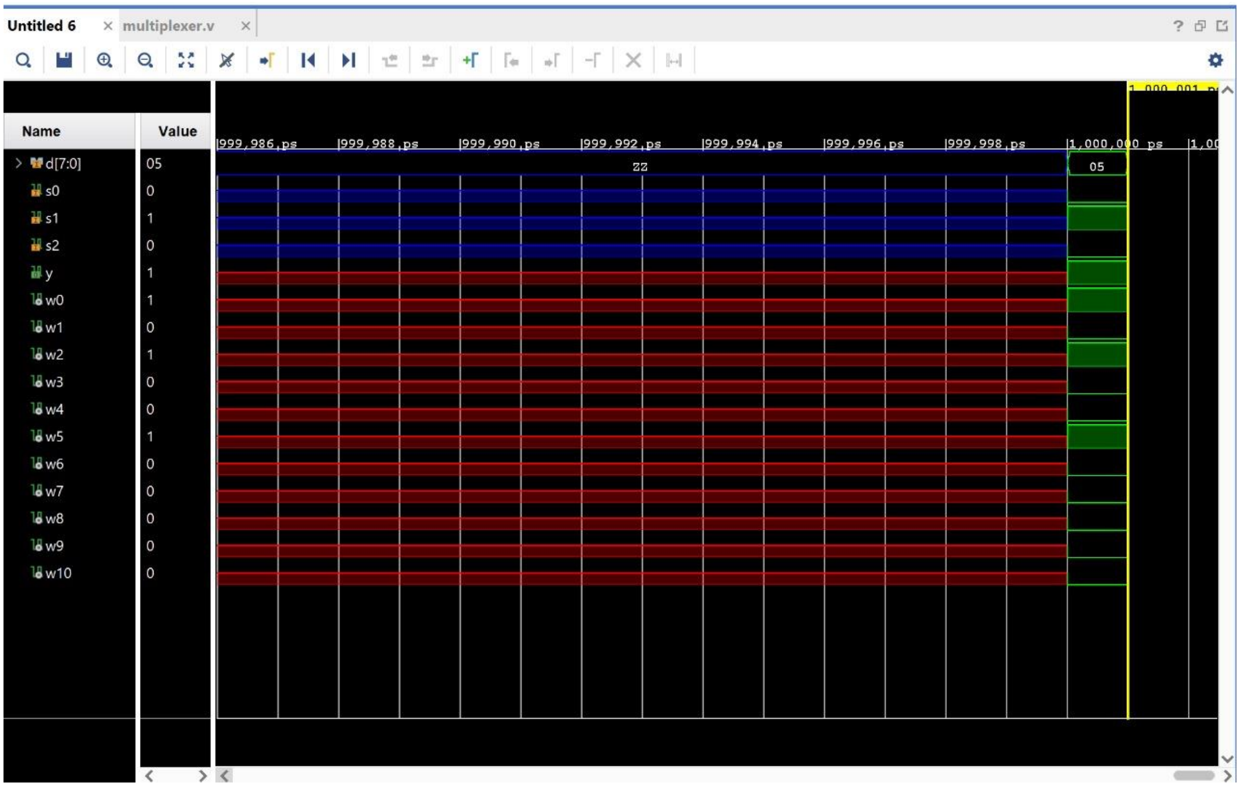
LOGIC DIAGRAM:



VERILOG CODE:

```
module multiplexer(s0,s1,s2,d,y);
input[7:0]d;
input s0,s1,s2;
output y;
wire w0,w1,w2,w3,w4,w5,w6,w7,w8,w9,w10;
not g1(w0,s0);
not g2(w1,s1);
not g3(w2,s2);
and g4(w3,d[0],w0,w1,w2);
and g5(w4,w0,d[1],w1,s2);
and g6(w5,d[2],w0,s1,w2);
and g7(w6,d[3],w0,s1,s2);
and g8(w7,d[4],w2,s0,s1);
and g9(w8,d[5],s0,w1,s2);
and g10(w9,d[6],s0,s1,w2);
and g11(w10,d[7],s0,s1,s2);
or g12(y,w3,w4,w5,w6,w7,w8,w9,w10);
endmodule
```

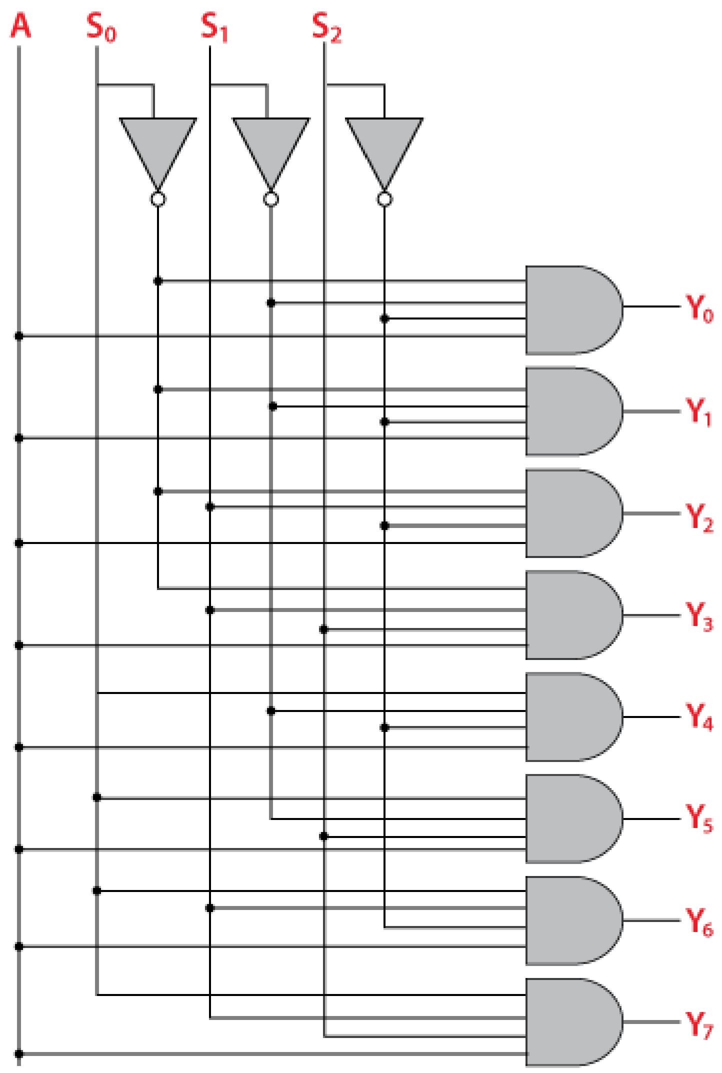
OUTPUT:



Name	Value
d[7:0]	05
s0	0
s1	1
s2	0
y	1
w0	1
w1	0
w2	1
w3	0
w4	0
w5	1
w6	0
w7	0
w8	0
w9	0
w10	0

DEMULTIPLEXER

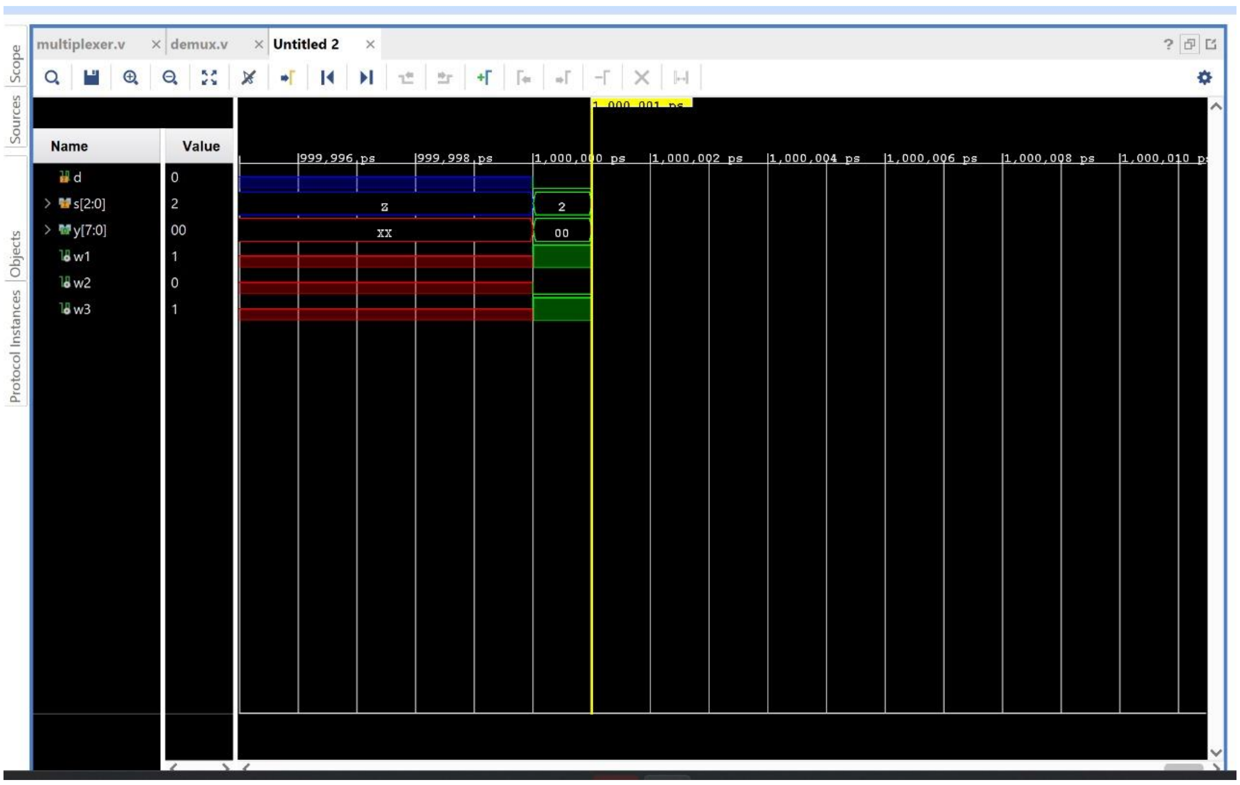
LOGIC DIAGRAM:



VERILOG CODE:

```
module demux(s,d,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
```

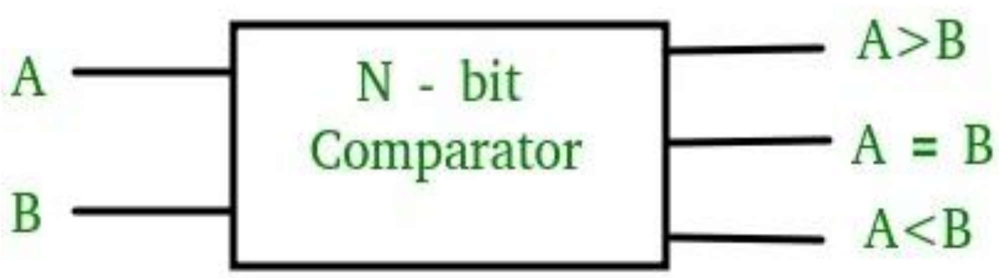
OUTPUT:



```
Name      Value
d         0
s[2:0]    2
y[7:0]    00
w1        1
w2        0
w3        1
```

MAGNITUDE COMPARATOR

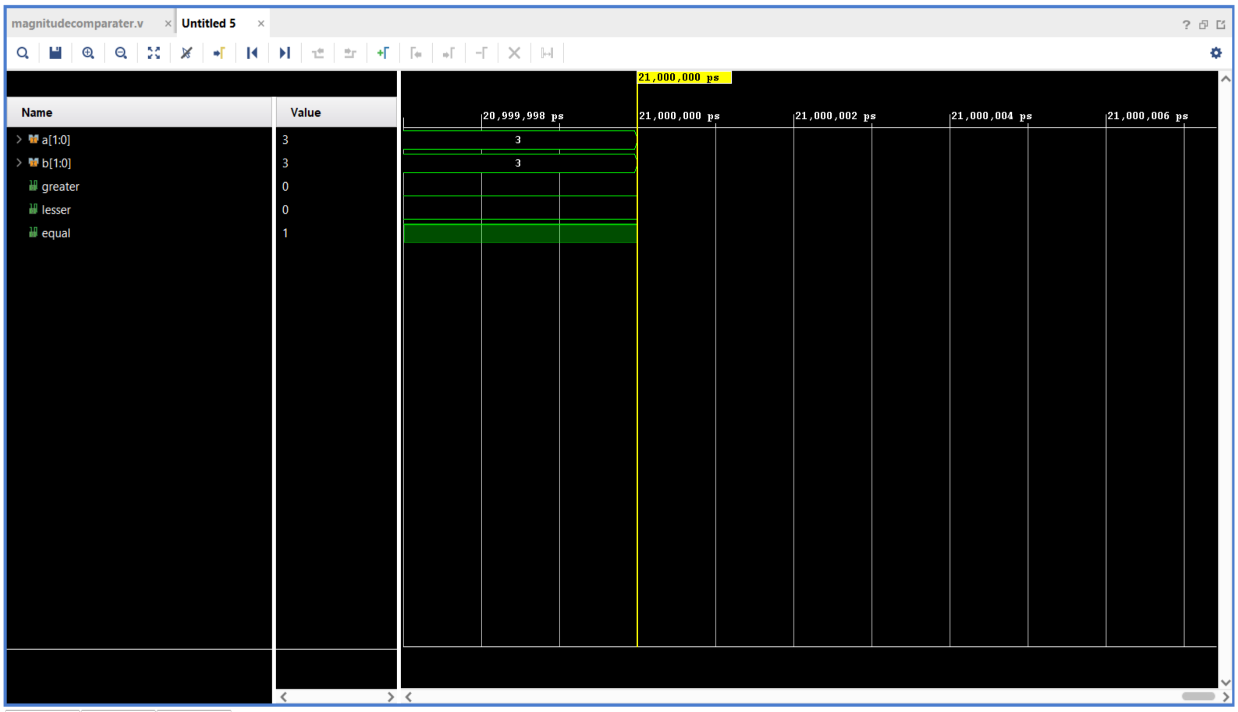
LOGIC DIAGRAM:



VERILOG CODE:

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

OUTPUT:



Name	Value
a[1:0]	3
b[1:0]	3
greater	0
lesser	0
equal	1

RESULT:

Thus the simulation and synthesis of ENCODER, DECODER, MULTIPLEXER, DEMULTIPLEXER, MAGNITUDE COMPARATOR using Xilinx ISE is simulated successfully.