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

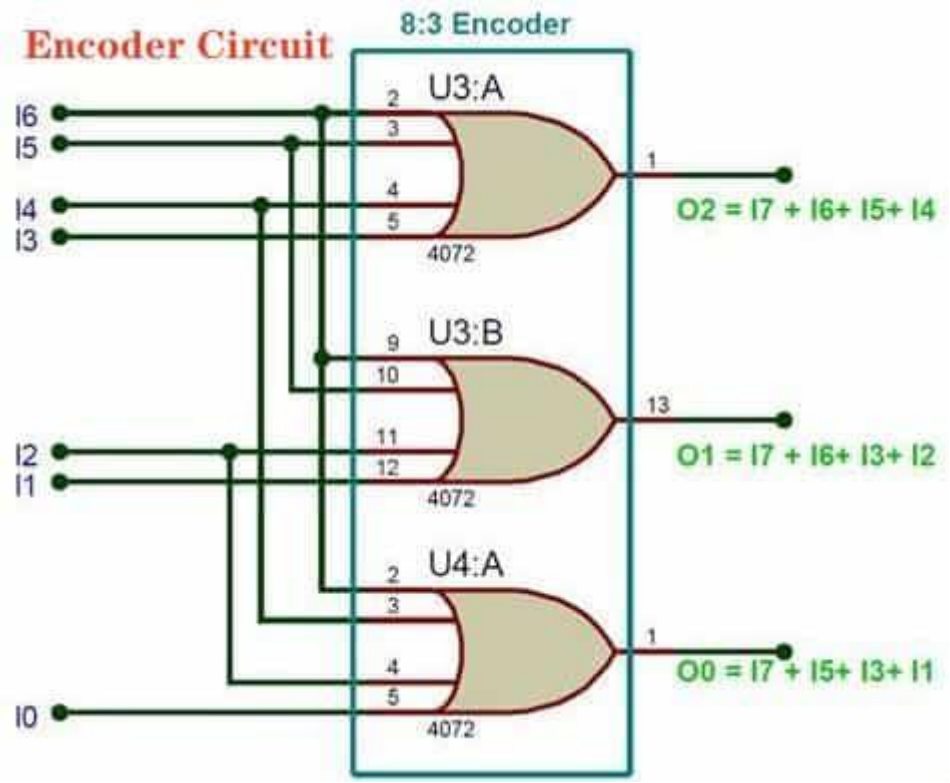
Double click on the Implement Design and double click on the Generate Programming File to create a bitstream of the design.(.v) file is converted into .bit file here.

STEP:11

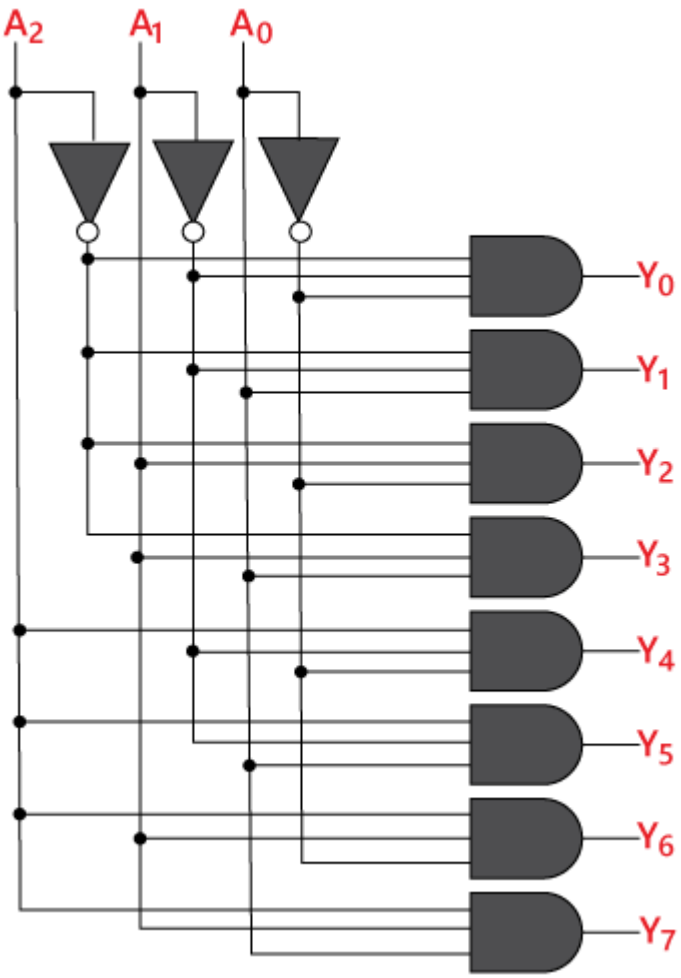
On the board, by giving required input, the LEDs starts to glow light, indicating the output.

LOGIC DIAGRAM:

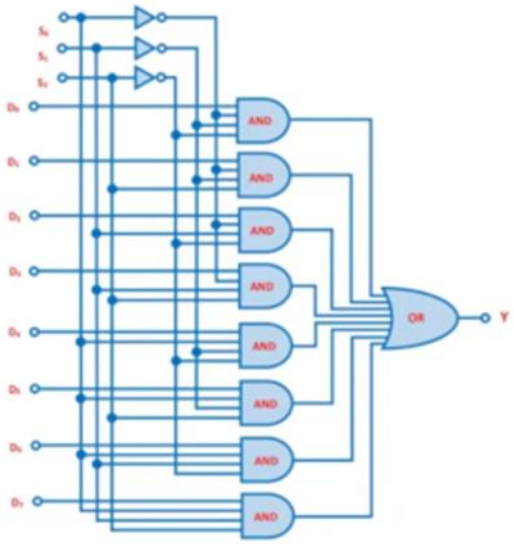
ENCODER:



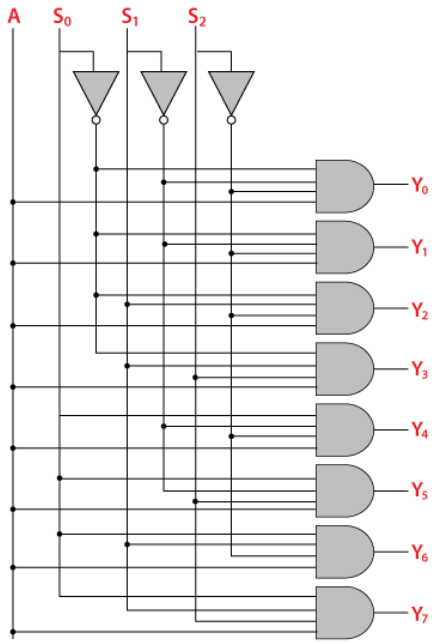
DECODER:



MULTIPLEXER:



DEMULTIPLEXER:



MAGNITUDE COMPARATOR:

VERILOG CODE:

ENCODER:

```
module encoder(i,o);

input [7:0]i;

output [2:0]o;

or g1(o[0],i[1],i[3],i[5],i[7]);

or g2(o[1],i[2],i[3],i[6],i[7]);

or g3(o[2],i[4],i[5],i[6],i[7]);

endmodule
```

DECODER:

```
module decoder(a,y);

input [2:0]a;

output [7:0]y;

wire w0,w1,w2;
```

```
not g1(w0,a[0]);

not g2(w1,a[1]);

not g3(w2,a[2]);

and g4(y[0],w0,w1,w2);

and g5(y[1],w2,w1,a[0]);

and g6(y[2],w2,a[1],w0);

and g7(y[3],w2,a[1],a[0]);

and g8(y[4],a[2],w1,w0);

and g9(y[5],a[2],w1,a[0]);

and g10(y[6],a[2],a[1],w0);

and g11(y[7],a[2],a[1],a[0]);

endmodule
```

MULTIPLEXER:

```
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
```

DE-MULTIPLEXER:

```
module demux8x1(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
```

MAGNITUDE COMPARATOR:

```
module magnitude(a,b,greater,lesser,equal);
input[2:0]a,b;
output reg greater,lesser,equal;
always @(*)
begin
if (a>b)
begin
greater=1'b1;
lesser=1'b0;
equal=1'b0;
end
else if(a<b)
begin
greater=1'b0;
lesser=1'b1;
equal=1'b0;
end
else
begin
greater=1'b0;
lesser=1'b0;
equal=1'b1;
```

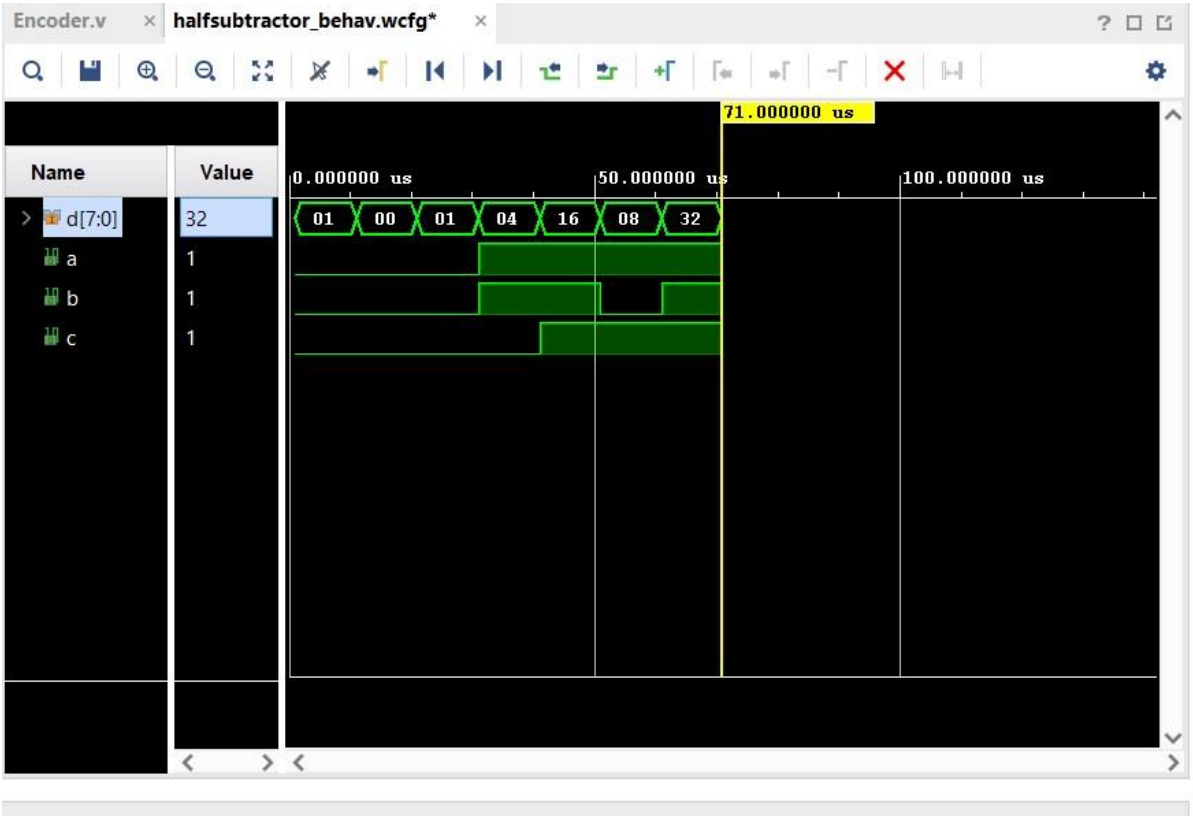
end

end

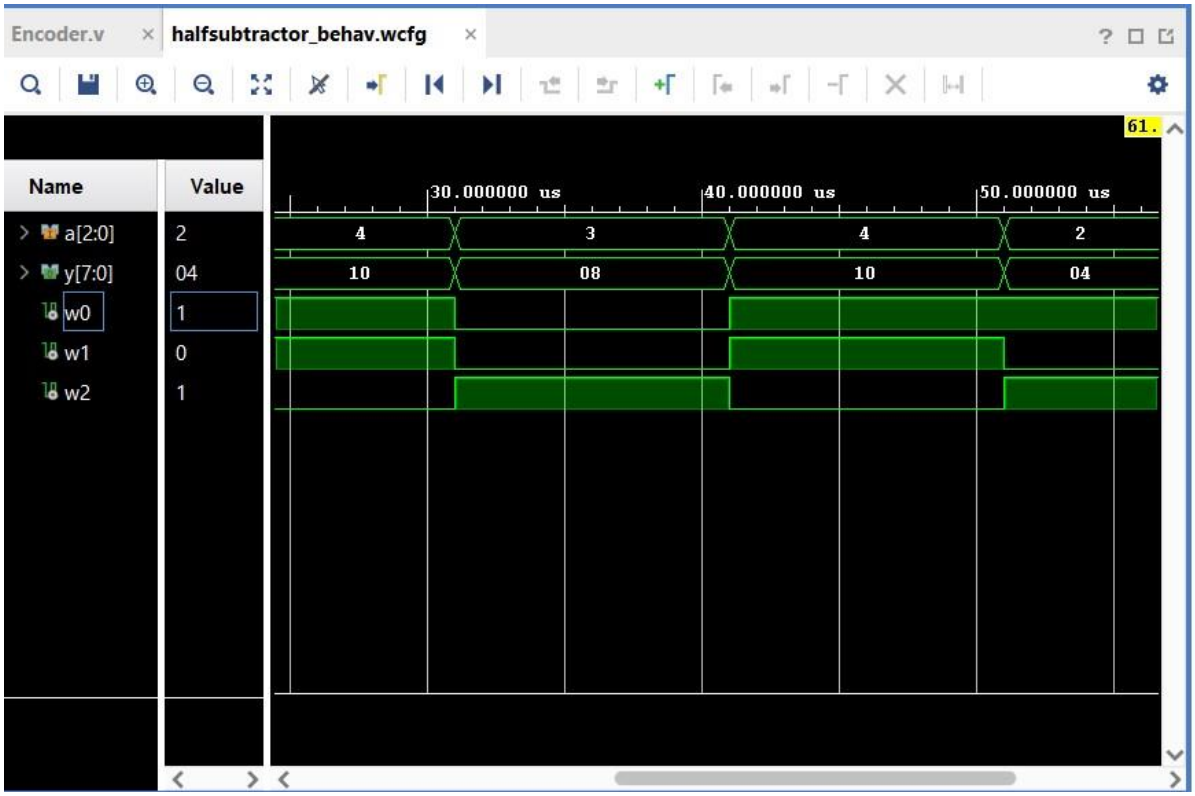
endmodule

OUTPUT WAVEFORM:

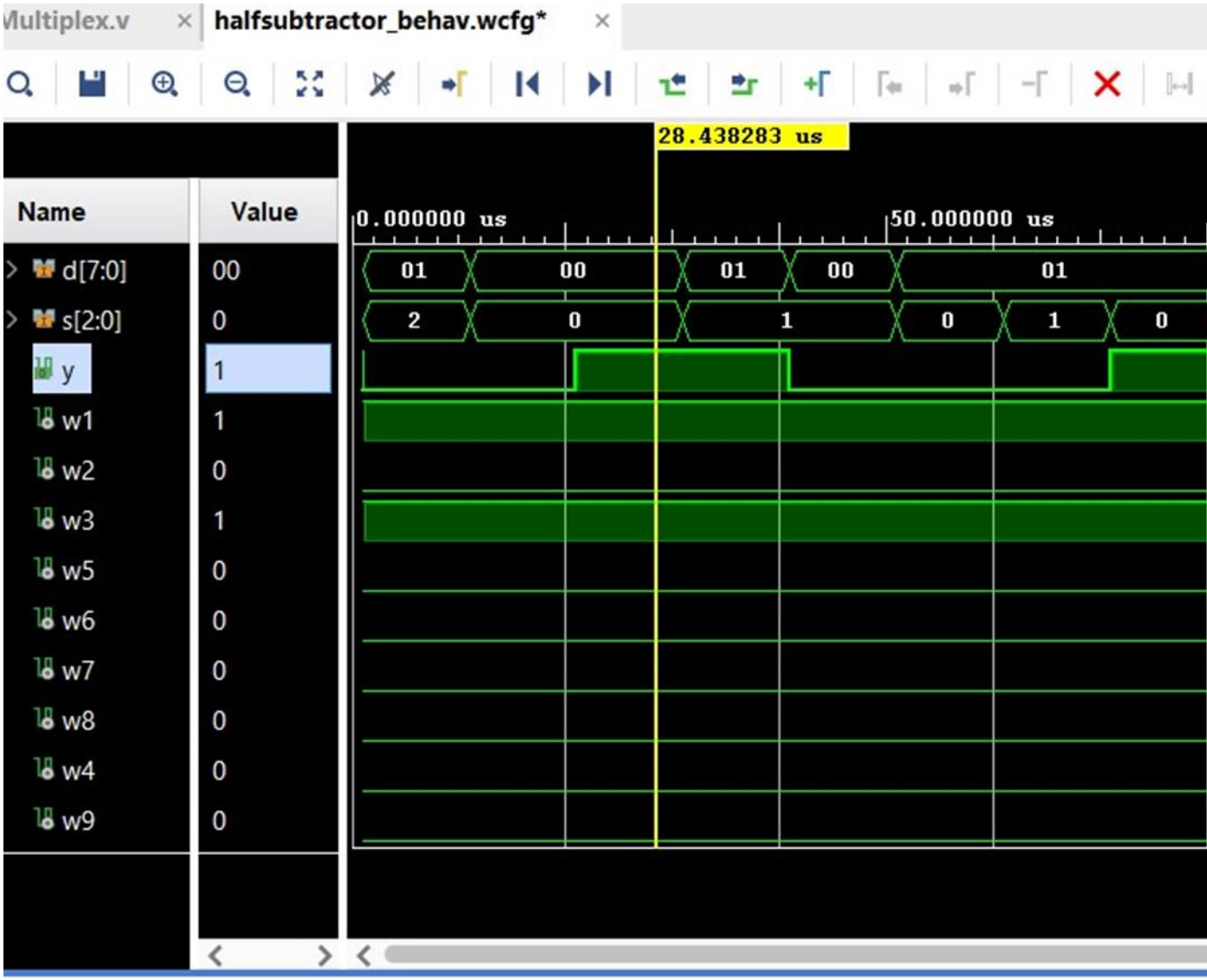
ENCODER:



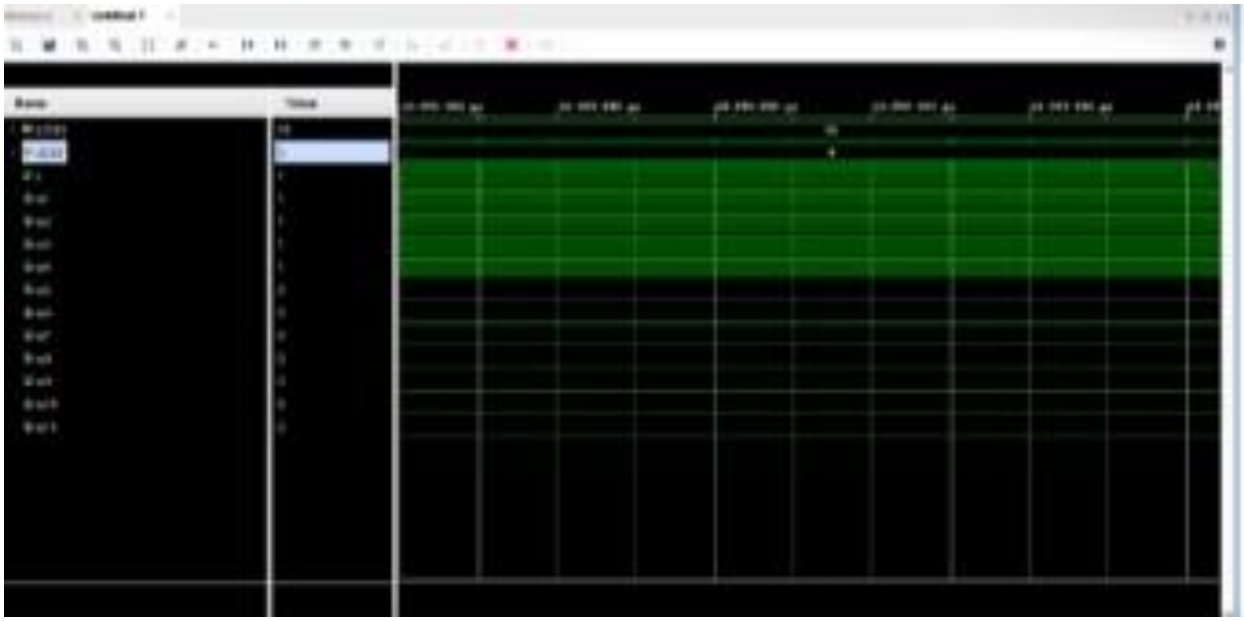
DECODER:


Encoder.v
halfsubtractor_behav.wcfg
Name
Value
a[2:0]
2
y[7:0]
04
w0
1
w1
0
w2
1
30.000000 us
40.000000 us
50.000000 us
4
3
4
2
10
08
10
04
61.

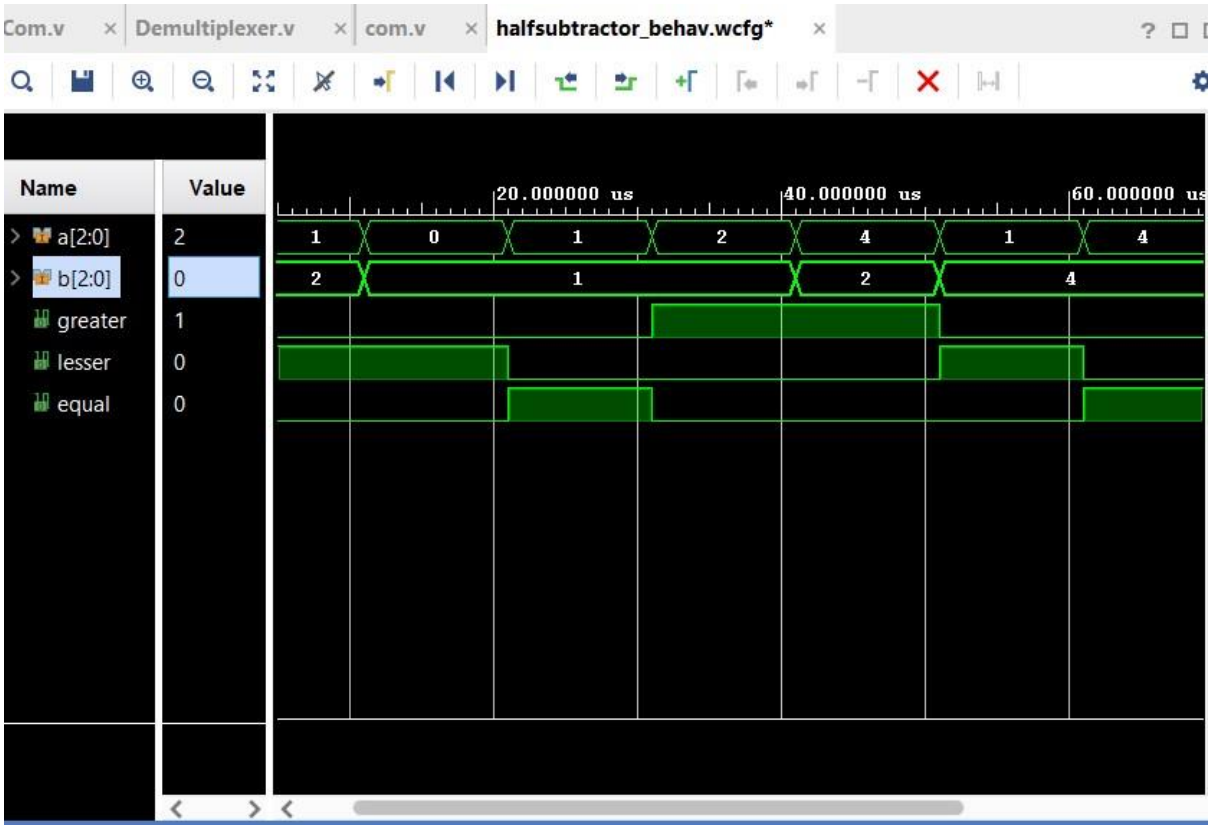
MULTIPLEXER:



DE-MULTIPLEXER:



MAGNITUDE COMPARATOR:



RESULT:

Hence, The simulation and synthesis Encoder, Decoder, Multiplexer, Demultiplexer, Magnitude Comparator was running successfully using Xilinx ISE.