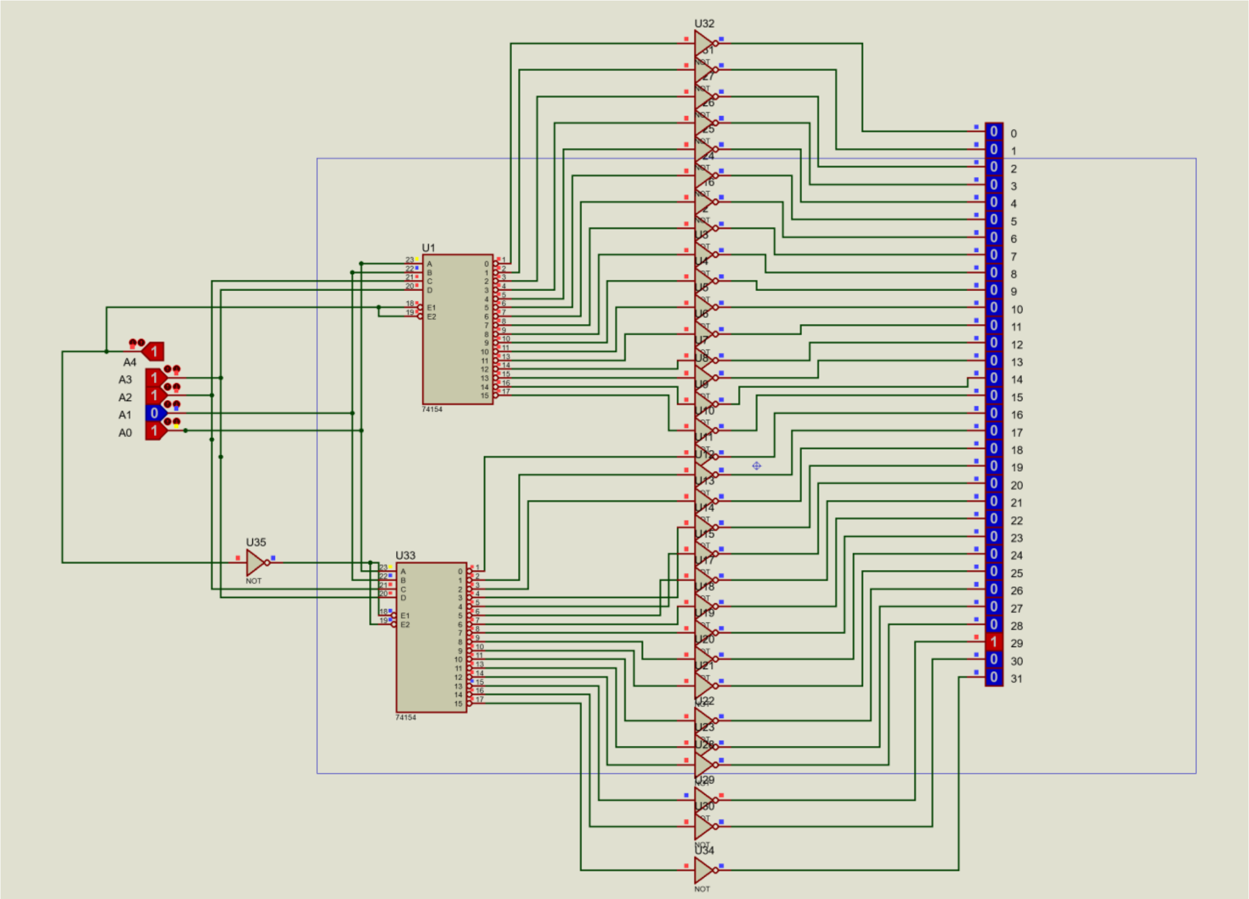
Report Task 1:



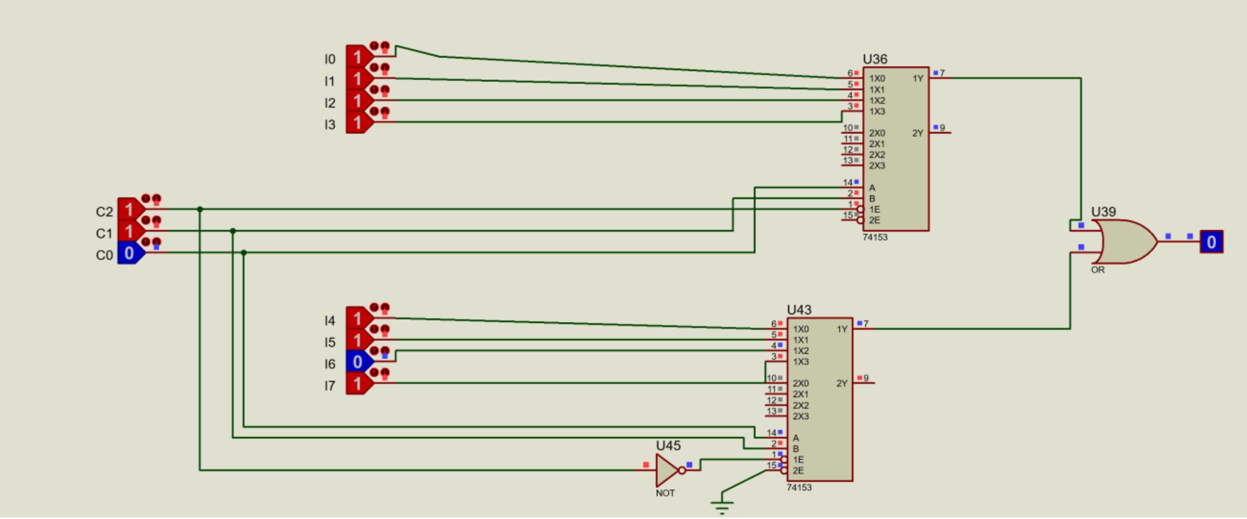
Here, we have used two 4 to 16 bit decoders to develop a 5 to 32 bit decoder. A4 was used to choose between U1 and U33.

Report Task 2

Truth Table

C2	C1	C0	Output
0	0	0	I0
0	0	1	I1
0	1	0	I2
0	1	1	I3
1	0	0	I4
1	0	1	I5
1	1	0	I6
1	1	1	I7

Simulated Circuit:



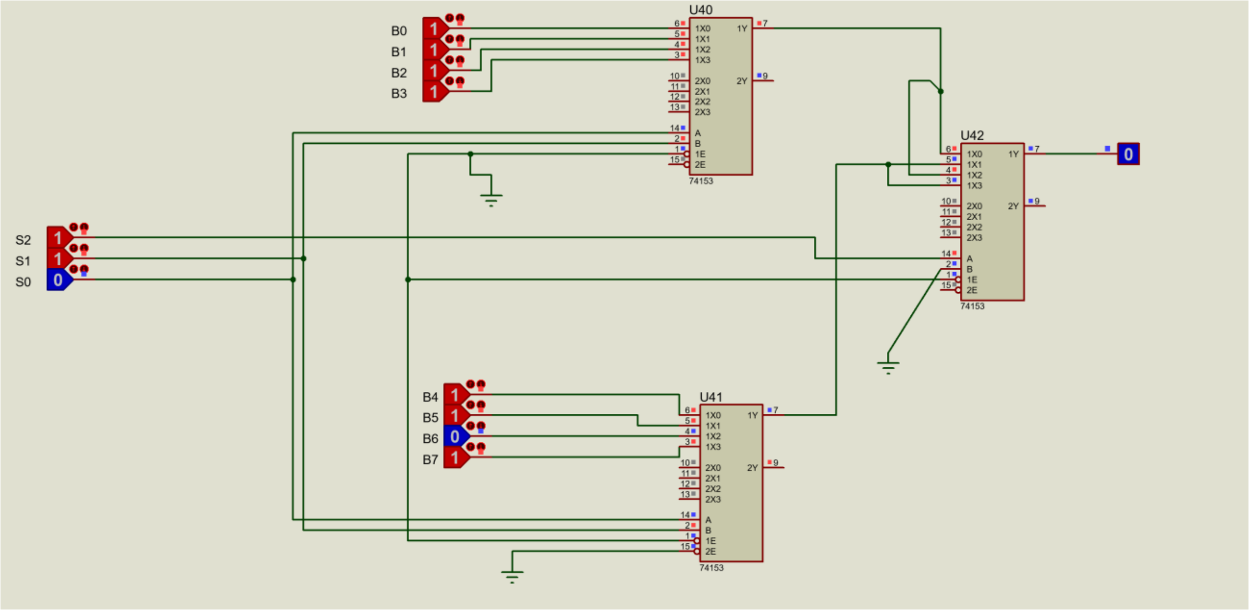
Here, the selector pin C2 is functioning as a turn off/on switch for either of the mux. For 110, we are getting the output from I6 pin.

Report Task 3

Truth Table

S2	S1	S0	Output
0	0	0	I0
0	0	1	I1
0	1	0	I2
0	1	1	I3
1	0	0	I4
1	0	1	I5
1	1	0	I6
1	1	1	I7

Simulated Circuit:



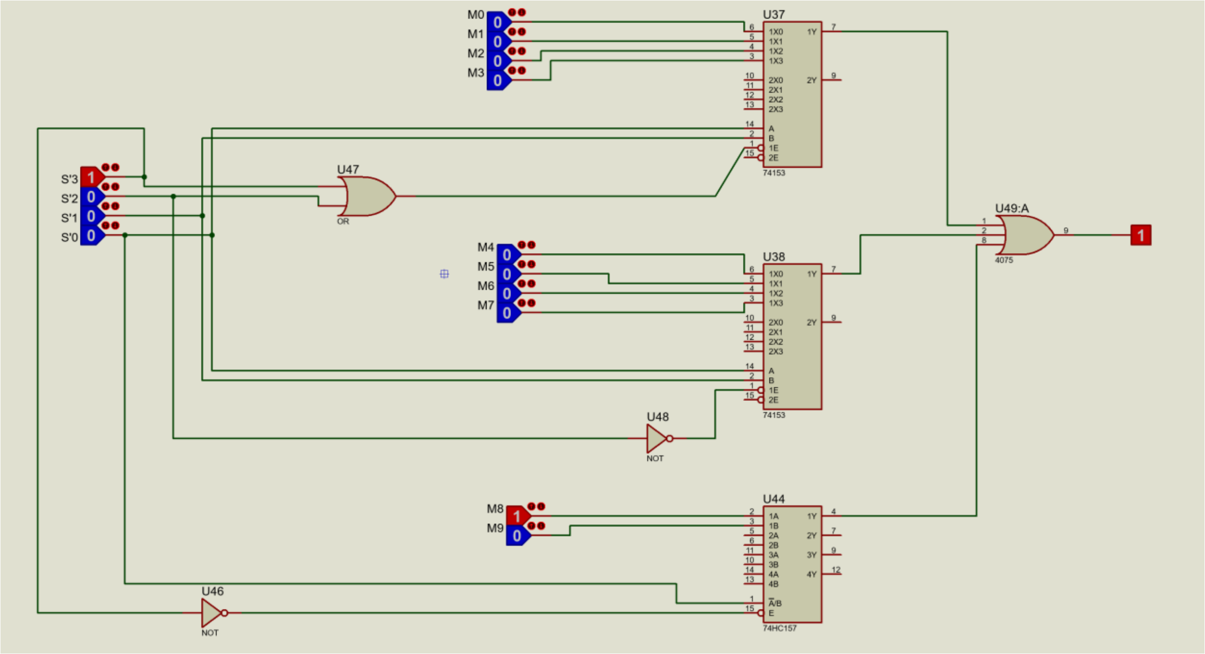
The selector bit S2 works in choosing the mux 4x1 between the U40 and U41 to function while choosing the output for the U42. The other selector bit works in choosing the output of the U40 and U41.

Report Task 4

Truth Table:

S'3	**S'2**	**S'1**	**S'0**	**Output**
0	**0**	**0**	**0**	**M0**
0	**0**	**0**	**1**	**M1**
0	**0**	**1**	**0**	**M2**
0	**0**	**1**	**1**	**M3**
0	**1**	**0**	**0**	**M4**
0	**1**	**0**	**1**	**M5**
0	**1**	**1**	**0**	**M6**
0	**1**	**1**	**1**	**M7**
1	**0**	**0**	**0**	**M8**
1	**0**	**0**	**1**	**M9**

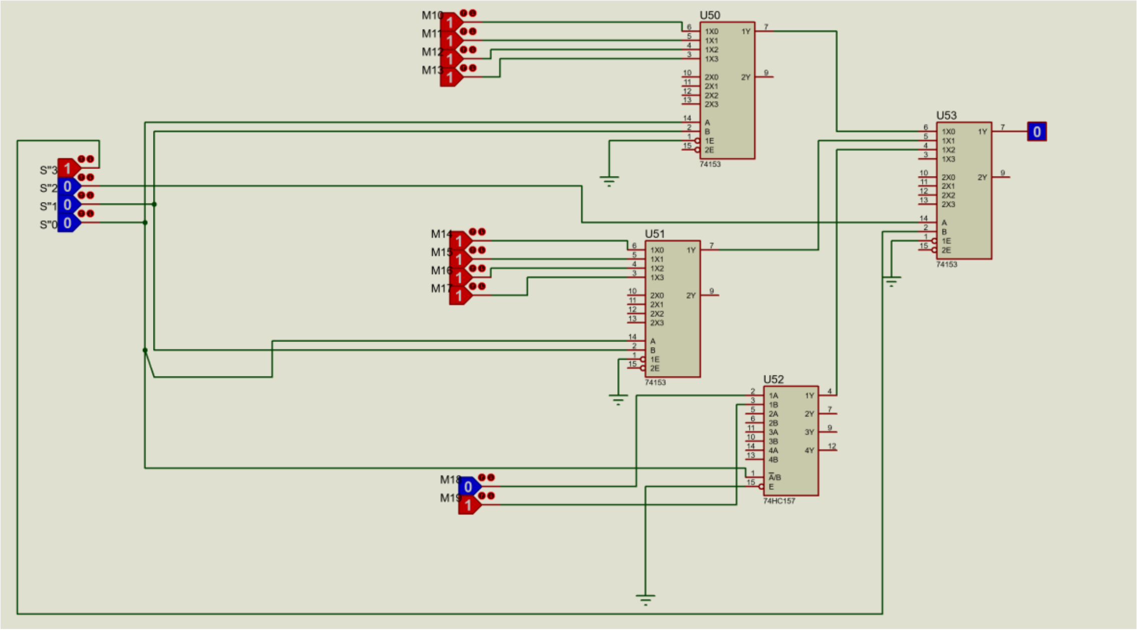
Simulated Circuit:



Here, we used S2 and S3 to turn on the required the mux as we need based on the truth table.

For example, selector bit for the 2x1 mux is S0 which happens to be the selector bit for other mux as well. But when S3 is 1 and S2 is zero, then only we need the 2x1 mux to be turned on and select among M8 and M9. For this purpose, one extra two-input OR gate was used at the left portion of the circuit.

Another way of Designing:



Report Task 5

Verilog Code

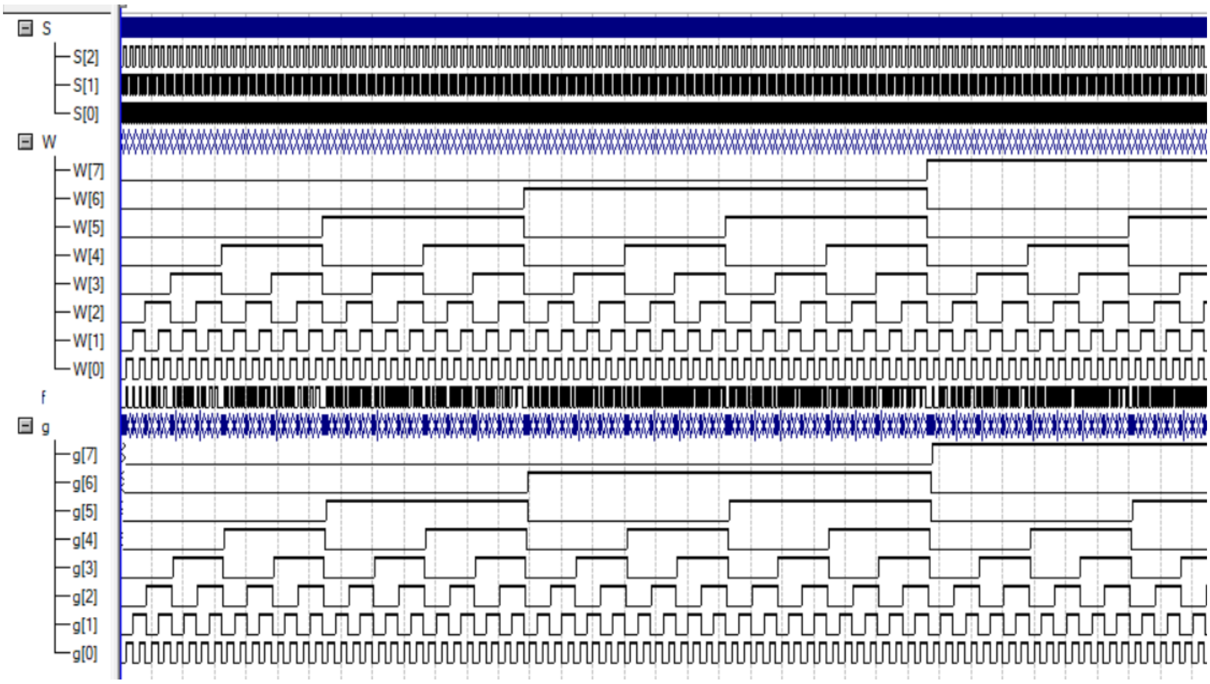
```
module task5(S,f,g,W,LED_COM);
input [2:0]S;
input [7:0]W;
output reg f;
output reg [7:0]g;
inout LED_COM;
assign LED_COM = 1;

//multiplexer circuit code
always @(S, W)
begin
if (S ==0) f = W[0];
else if (S ==1) f = W[1];
else if (S ==2) f = W[2];
else if (S ==3) f = W[3];
else if (S ==4) f = W[4];
else if (S ==5) f = W[5];
else if (S==6) f = W[6];
else if (S==7) f = W[7];
end

//demultiplexer circuit code
always @(f)
begin
```

```
if (S==0) g[0] = f;
else if (S==1) g[1] = f;
else if (S==2) g[2] = f;
else if (S==3) g[3] = f;
else if (S==4) g[4] = f;
else if (S==5) g[5] = f;
else if (S==6) g[6] = f;
else if (S==7) g[7] = f;
end
endmodule
```

Vector Waveform:


S
S[2]
S[1]
S[0]
W
W[7]
W[6]
W[5]
W[4]
W[3]
W[2]
W[1]
W[0]
f
g
g[7]
g[6]
g[5]
g[4]
g[3]
g[2]
g[1]
g[0]