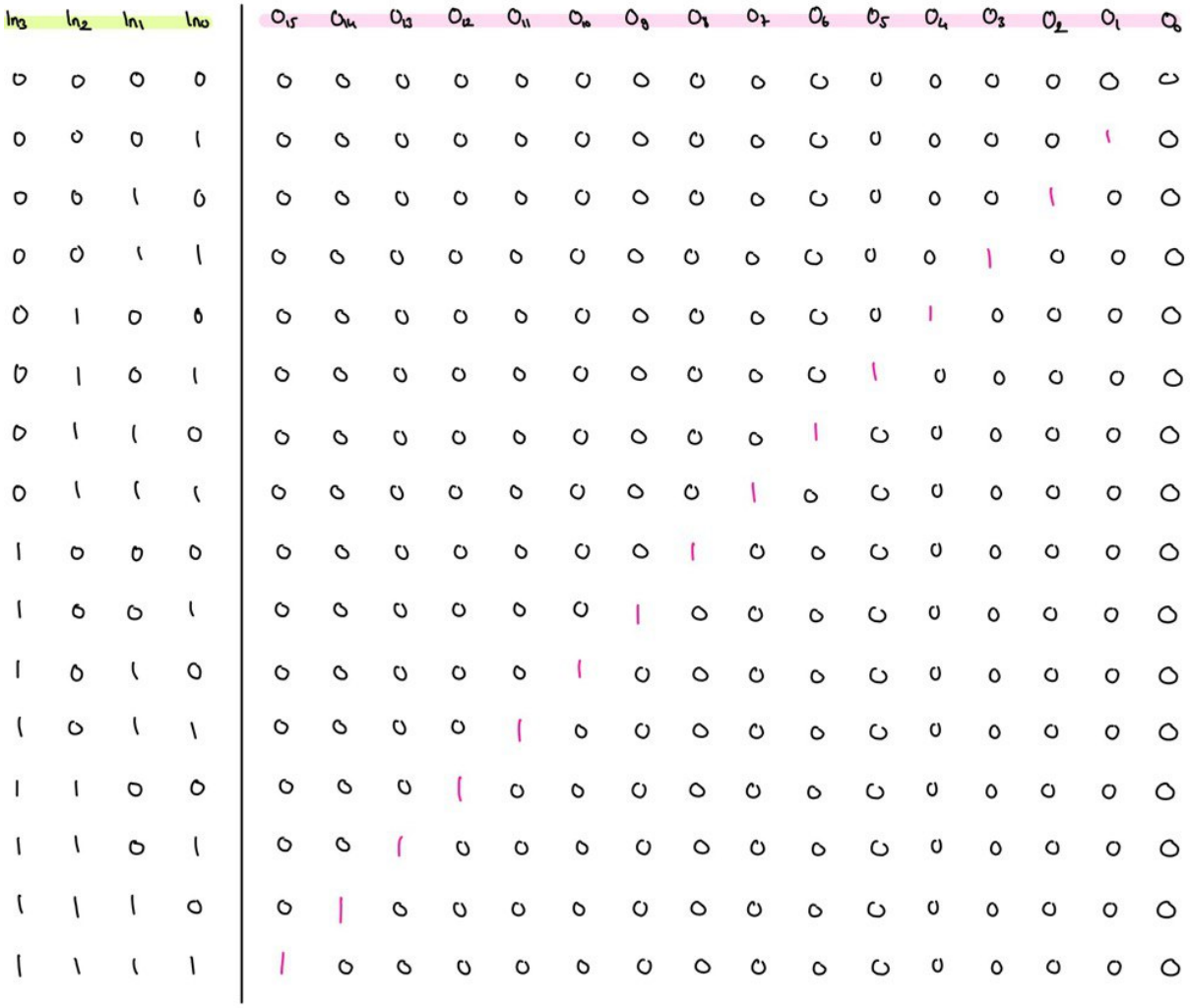
Decoder

4 bit input 16 bit output decoder truth table:



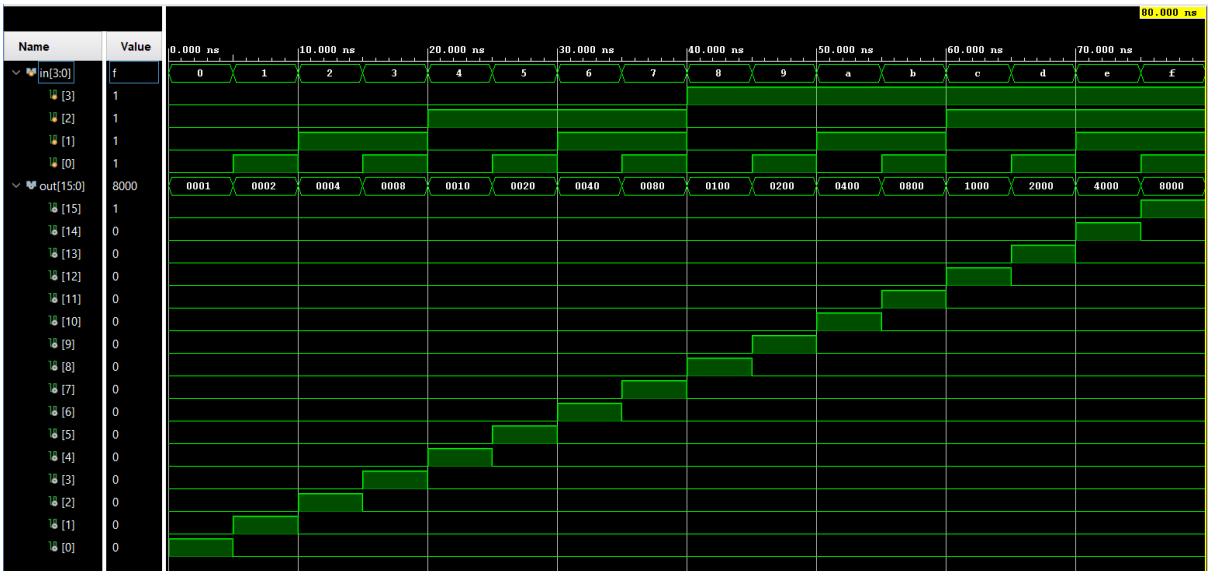
Decoder vivado code and testbench code:

```
`timescale 1ns / 1ps
module DECODER(
input [3:0] IN,
output reg [15:0] OUT
);
always @(IN) begin
    case(IN)
    4'd0    :   OUT = 16'b0000000000000001;
    4'd1    :   OUT = 16'b0000000000000010;
    4'd2    :   OUT = 16'b0000000000000100;
    4'd3    :   OUT = 16'b0000000000001000;
    4'd4    :   OUT = 16'b0000000000010000;
    4'd5    :   OUT = 16'b0000000000100000;
    4'd6    :   OUT = 16'b0000000001000000;
    4'd7    :   OUT = 16'b0000000010000000;
    4'd8    :   OUT = 16'b0000000100000000;
    4'd9    :   OUT = 16'b0000001000000000;
    4'd10    :   OUT = 16'b0000010000000000;
    4'd11    :   OUT = 16'b0000100000000000;
    4'd12    :   OUT = 16'b0001000000000000;
    4'd13    :   OUT = 16'b0010000000000000;
    4'd14    :   OUT = 16'b0100000000000000;
    4'd15    :   OUT = 16'b1000000000000000;

    default :   OUT=16'b0;
    endcase
end
endmodule
```

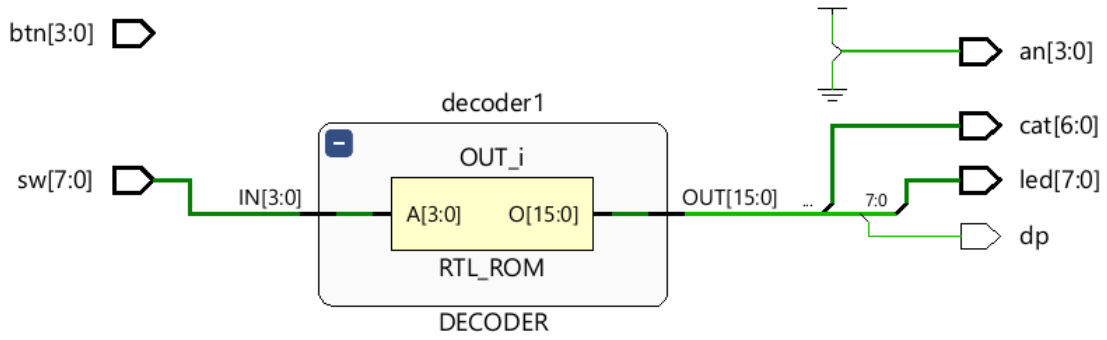
```
module DECODER_tb;

reg [3:0] in;
wire [15:0] out;

Top_Module dut(.sw(in), .led(out[7:0]), .cat(out[14:8]), .dp(out[15]));

always
begin
```

```
#5 in[0]=~in[0];
end
always
begin
#10 in[1]=~in[1];
end
always
begin
#20 in[2]=~in[2];
end
always
begin
#40 in[3]=~in[3];
end

initial
begin
in=0;
#80
$finish;
end
endmodule
```

Simulation wave:
all of the possible input combinations are obtained, and the output came as expected.

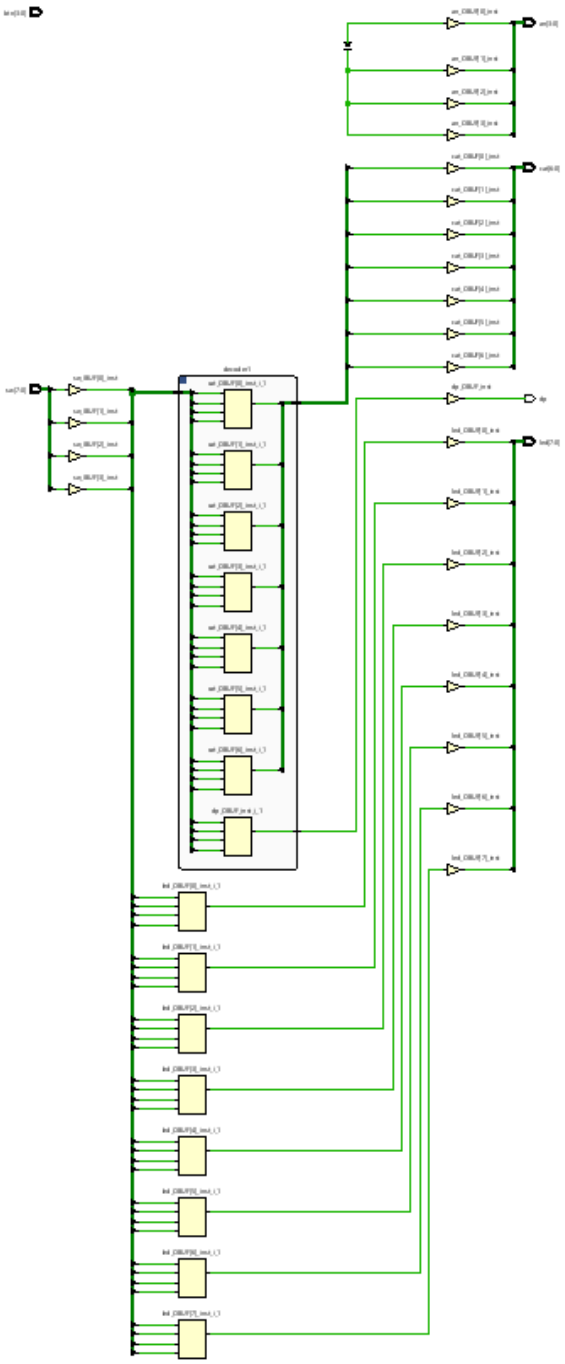


RTL Schematic for Decoder:



Technology Schematic for Decoder:

There are 16 LUT in technology schematic, every luts logic statement is different from each other because they give 1 at the output at different conditions. OBUF[i] is 1 when the input is i.



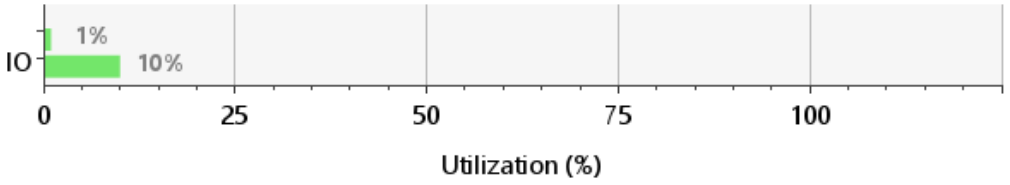
Cell Properties

OUT_OBUF[13]_inst_i_1

I3	I2	I1	I0	O=I0 & I1 & !I2 & I3
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Summary

Resource	Utilization	Available	Utilization %
LUT	8	32600	0.02
IO	20	210	9.52



The maximum delay is 10.448 before, and after the constraint has changed the maximum path delay reduced to 9.643

combinational delays before:

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
sw[0]	cat[1]			3.405	FAST
sw[0]	cat[0]	10.342	SLOW	3.338	FAST
sw[1]	cat[1]	10.296	SLOW	3.283	FAST
sw[0]	led[6]	10.126	SLOW	3.232	FAST
sw[1]	cat[0]	10.076	SLOW	3.216	FAST
sw[0]	led[7]	9.960	SLOW	3.152	FAST
sw[0]	led[5]	9.919	SLOW	3.171	FAST
sw[1]	led[7]	9.899	SLOW	3.114	FAST
sw[3]	led[0]	9.858	SLOW	3.107	FAST
sw[1]	led[6]	9.850	SLOW	3.098	FAST
sw[3]	cat[4]	9.832	SLOW	3.100	FAST
sw[1]	cat[5]	9.675	SLOW	3.035	FAST
sw[1]	led[5]	9.642	SLOW	3.036	FAST
sw[0]	cat[5]	9.634	SLOW	3.022	FAST
sw[3]	led[2]	9.551	SLOW	3.021	FAST
sw[3]	dp	9.395	SLOW	2.968	FAST
sw[2]	led[2]	9.373	SLOW	2.864	FAST
sw[1]	led[2]	9.342	SLOW	2.817	FAST
sw[0]	led[0]	9.304	SLOW	2.860	FAST
sw[0]	cat[4]	9.287	SLOW	2.866	FAST
sw[3]	cat[5]	9.247	SLOW	2.914	FAST
sw[2]	dp	9.229	SLOW	2.824	FAST
sw[3]	cat[3]	9.224	SLOW	2.918	FAST
sw[1]	dp	9.199	SLOW	2.776	FAST
sw[3]	led[7]	9.186	SLOW	2.831	FAST
sw[2]	cat[1]	9.178	SLOW	2.868	FAST
sw[3]	cat[2]	9.144	SLOW	2.828	FAST
sw[0]	led[3]	9.076	SLOW	2.796	FAST
sw[3]	cat[1]	9.045	SLOW	2.716	FAST
sw[1]	cat[4]	9.035	SLOW	2.731	FAST
sw[3]	led[4]	9.034	SLOW	2.807	FAST
sw[2]	cat[3]	9.025	SLOW	2.779	FAST
sw[1]	led[3]	9.014	SLOW	2.758	FAST
sw[1]	cat[3]	8.993	SLOW	2.729	FAST
sw[2]	cat[0]	8.992	SLOW	2.797	FAST
sw[1]	cat[6]	8.945	SLOW	2.695	FAST
sw[2]	cat[5]	8.908	SLOW	2.718	FAST
sw[0]	cat[6]	8.903	SLOW	2.683	FAST
sw[1]	led[0]	8.857	SLOW	2.664	FAST
sw[3]	cat[0]	8.827	SLOW	2.647	FAST
sw[0]	led[2]	8.822	SLOW	2.624	FAST
sw[3]	led[1]	8.806	SLOW	2.728	FAST
sw[2]	led[6]	8.744	SLOW	2.695	FAST
sw[2]	led[0]	8.722	SLOW	2.654	FAST
sw[2]	led[7]	8.715	SLOW	2.668	FAST
sw[2]	cat[4]	8.694	SLOW	2.648	FAST
sw[0]	dp	8.683	SLOW	2.583	FAST
sw[0]	cat[2]	8.600	SLOW	2.595	FAST
sw[2]	led[1]	8.595	SLOW	2.578	FAST
sw[2]	led[5]	8.569	SLOW	2.628	FAST
sw[1]	led[1]	8.564	SLOW	2.528	FAST
sw[3]	cat[6]	8.483	SLOW	2.578	FAST
sw[0]	led[4]	8.479	SLOW	2.560	FAST
sw[0]	cat[3]	8.479	SLOW	2.535	FAST
sw[3]	led[6]	8.406	SLOW	2.469	FAST

sw[3]	led[6]	8.406	SLOW	2.469	FAST
sw[1]	cat[2]	8.348	SLOW	2.456	FAST
sw[3]	led[3]	8.304	SLOW	2.477	FAST
sw[3]	led[5]	8.200	SLOW	2.407	FAST
sw[2]	cat[6]	8.175	SLOW	2.377	FAST
sw[0]	led[1]	8.046	SLOW	2.335	FAST
sw[2]	cat[2]	8.039	SLOW	2.371	FAST
sw[1]	led[4]	8.035	SLOW	2.362	FAST
sw[2]	led[4]	7.930	SLOW	2.348	FAST
sw[2]	led[3]	7.864	SLOW	2.308	FAST

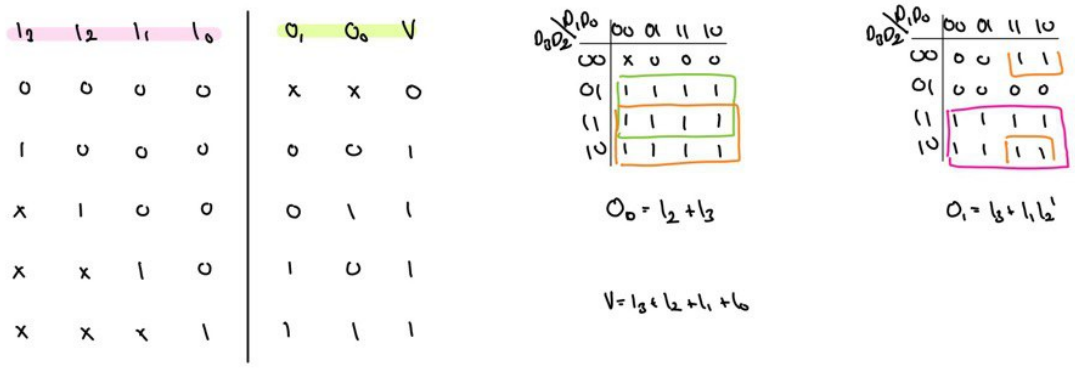
Combinational delays after

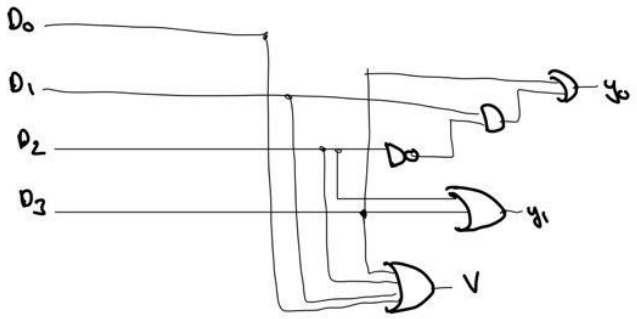
From Port	To Port	Ma 1	Max Process Corner
sw[3]	led[2]	9.643	SLOW
sw[0]	led[2]	9.488	SLOW
sw[3]	cat[4]	9.462	SLOW
sw[3]	led[0]	9.438	SLOW
sw[0]	cat[5]	9.414	SLOW
sw[3]	cat[3]	9.388	SLOW
sw[2]	cat[0]	9.363	SLOW
sw[0]	cat[1]	9.323	SLOW
sw[2]	led[2]	9.307	SLOW
sw[0]	cat[0]	9.302	SLOW
sw[2]	cat[1]	9.239	SLOW
sw[2]	led[0]	9.211	SLOW
sw[3]	cat[2]	9.210	SLOW
sw[1]	cat[5]	9.209	SLOW
sw[1]	cat[1]	9.202	SLOW
sw[2]	cat[5]	9.193	SLOW
sw[3]	led[7]	9.193	SLOW
sw[0]	led[6]	9.192	SLOW
sw[3]	led[6]	9.191	SLOW
sw[3]	dp	9.188	SLOW
sw[0]	led[7]	9.175	SLOW
sw[1]	led[6]	9.151	SLOW
sw[0]	cat[6]	9.149	SLOW
sw[1]	led[0]	9.107	SLOW
sw[3]	led[5]	9.096	SLOW
sw[1]	cat[0]	9.087	SLOW
sw[0]	led[0]	9.082	SLOW
sw[0]	led[0]	9.082	SLOW
sw[3]	cat[0]	9.080	SLOW
sw[0]	led[5]	9.056	SLOW
sw[2]	cat[3]	9.054	SLOW
sw[3]	cat[5]	8.976	SLOW
sw[0]	cat[4]	8.963	SLOW
sw[1]	led[7]	8.959	SLOW
sw[1]	cat[6]	8.944	SLOW
sw[2]	cat[6]	8.930	SLOW
sw[2]	dp	8.901	SLOW
sw[1]	dp	8.880	SLOW
sw[1]	cat[4]	8.877	SLOW
sw[1]	led[2]	8.840	SLOW
sw[3]	led[1]	8.823	SLOW
sw[2]	led[7]	8.783	SLOW
sw[3]	led[4]	8.776	SLOW
sw[2]	led[6]	8.772	SLOW
sw[0]	cat[3]	8.761	SLOW
sw[0]	cat[2]	8.745	SLOW
sw[2]	led[5]	8.719	SLOW
sw[3]	cat[1]	8.719	SLOW
sw[2]	cat[4]	8.710	SLOW
sw[3]	cat[6]	8.679	SLOW
sw[0]	led[1]	8.665	SLOW
sw[3]	led[3]	8.642	SLOW
sw[1]	led[5]	8.638	SLOW
sw[1]	cat[3]	8.632	SLOW

sw[1]	cat[3]	8.632	SLOW
sw[1]	cat[2]	8.627	SLOW
sw[0]	led[3]	8.600	SLOW
sw[2]	led[4]	8.550	SLOW
sw[2]	led[1]	8.486	SLOW
sw[2]	cat[2]	8.460	SLOW
sw[0]	led[4]	8.453	SLOW
sw[1]	led[4]	8.443	SLOW
sw[0]	dp	8.419	SLOW
sw[2]	led[3]	8.275	SLOW
sw[1]	led[3]	8.190	SLOW
sw[1]	led[1]	8.051	SLOW

Encoder

Truth table and Karnaugh maps for Encoder:





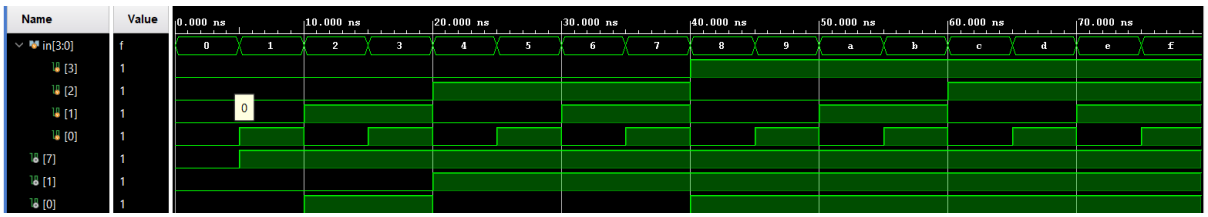
vivado code and testbench code for encoder1

```
module ENCODER(

input [3:0] IN,

output [1:0] OUT,

output V

);
```

```
wire fo1, fo2, fo3;

assign V = IN[0] | IN[1] | IN[2] | IN[3];
assign OUT[1] = IN[2] | IN[3];
assign OUT[0] = (~IN[2] & IN[1]) | IN[3];
endmodule
```

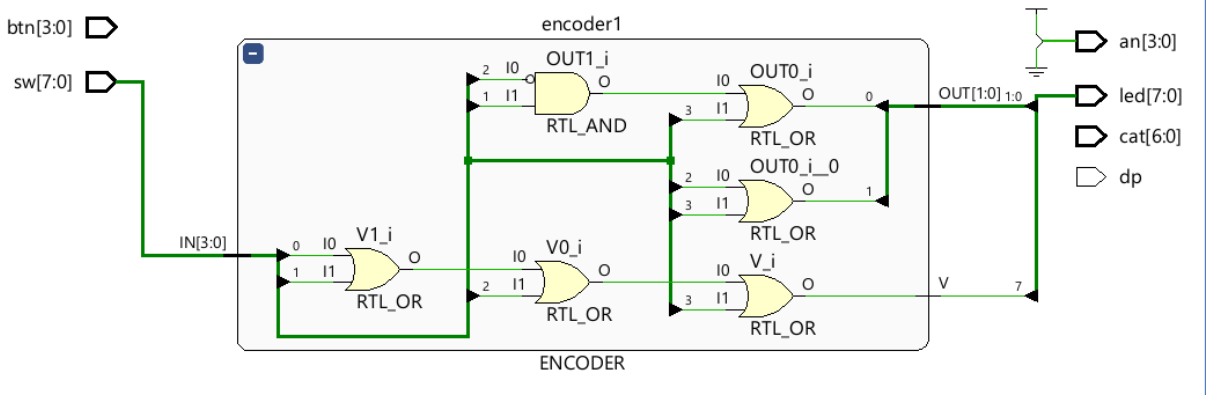
```
module ENCODER_tb;
reg [3:0] in;
wire [7:0] out;

Top_Module dut(.sw(in), .led(out));

always
begin
#5 in[0]=~in[0];
end
always
begin
#10 in[1]=~in[1];
end
always
begin
#20 in[2]=~in[2];
end
always
begin
#40 in[3]=~in[3];
end
initial
begin
in=0;
#80
$finish;
```

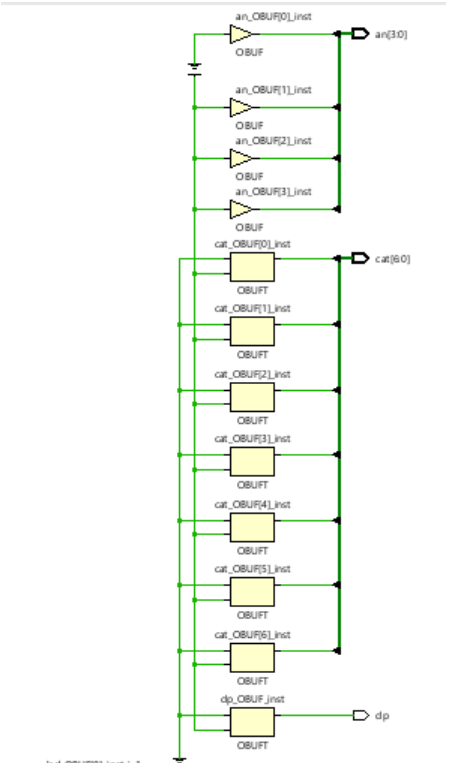
Simulation wave:
all of the possible input combinations are obtained, and the output came as expected.

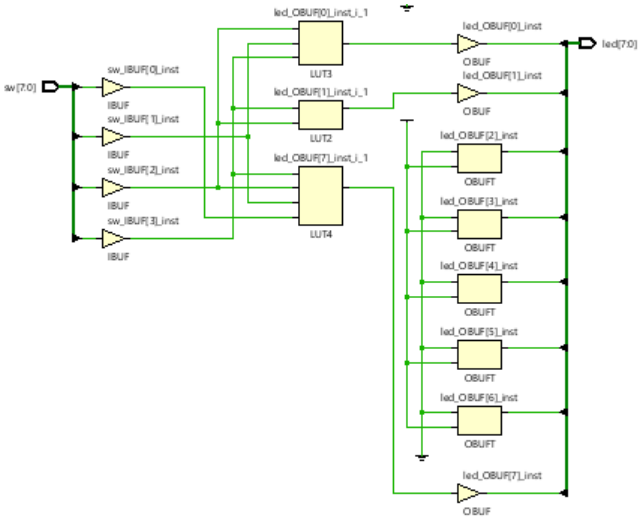


RTL and Technology Schematics for Encoder1

For the encoder part of the module the RTL schematic has 6 gates but the Technology schematic has 8 LUTs.



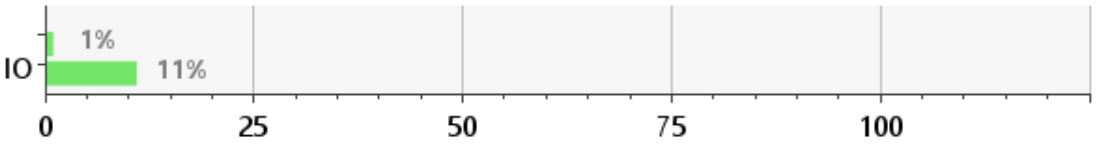




Timing and Utilization report before implementation:

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
sw[0]	led[7]	6.746	SLOW	2.219	FAST
sw[1]	led[0]	6.732	SLOW	2.196	FAST
sw[1]	led[7]	6.771	SLOW	2.211	FAST
sw[2]	led[0]	6.718	SLOW	2.182	FAST
sw[2]	led[1]	6.736	SLOW	2.201	FAST
sw[2]	led[7]	6.757	SLOW	2.192	FAST
sw[3]	led[0]	6.716	SLOW	2.181	FAST
sw[3]	led[1]	6.735	SLOW	2.200	FAST
sw[3]	led[7]	6.758	SLOW	2.193	FAST

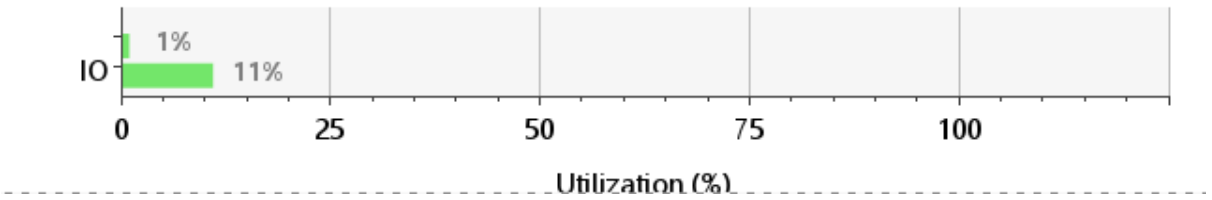
Resource	Utilization	Available	Utilization %
LUT	2	32600	0.01
IO	24	210	11.43



Timing and Utilization Report after implementation

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
sw[0]	led[7]	9.458	SLOW	2.936	FAST
sw[1]	led[0]	9.074	SLOW	2.704	FAST
sw[1]	led[7]	9.262	SLOW	2.831	FAST
sw[2]	led[0]	9.085	SLOW	2.729	FAST
sw[2]	led[1]	8.456	SLOW	2.497	FAST
sw[2]	led[7]	9.275	SLOW	2.859	FAST
sw[3]	led[0]	9.251	SLOW	2.867	FAST
sw[3]	led[1]	8.666	SLOW	2.641	FAST
sw[3]	led[7]	9.473	SLOW	2.991	FAST

Resource	Utilization	Available	Utilization %
LUT	2	32600	0.01
IO	24	210	11.43

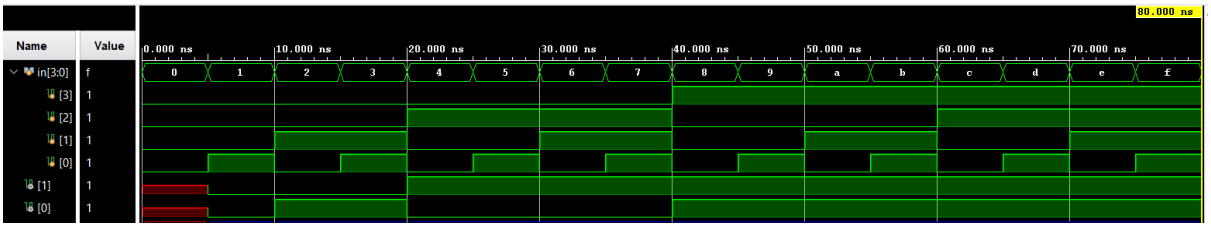


Encoder2

Code for encoder with case structure:

```
module ENCODER2(
input [3:0] IN,
output reg  [1:0] OUT,
output reg V );

always @(IN) begin
    casex(IN)
    4'b0000 : begin  OUT=2'bXX; V=1'b0; end
    4'b0001 : begin  OUT=2'b00; V=1'b1; end
    4'b001x : begin  OUT=2'b01; V=1'b1; end
    4'b01xx : begin  OUT=2'b10; V=1'b1; end
    4'b1xxx : begin  OUT=2'b11; V=1'b1; end
    default : OUT=2'd0;
    endcase
end

endmodule
```

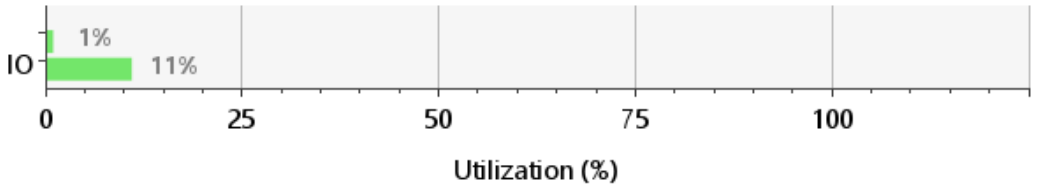
Simulation wave:
all of the possible input combinations are obtained, and the output came as expected.



Timing and Utilization Report:

The utilization report does not changed and the maximum path delay changed a little.

Resource	Utilization	Available	Utilization %
LUT	2	32600	0.01
IO	24	210	11.43



From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
sw[0]	led[7]	9.288	SLOW	2.881	FAST
sw[1]	led[0]	9.058	SLOW	2.711	FAST
sw[1]	led[7]	9.091	SLOW	2.772	FAST
sw[2]	led[0]	9.085	SLOW	2.729	FAST
sw[2]	led[1]	8.455	SLOW	2.517	FAST
sw[2]	led[7]	9.094	SLOW	2.789	FAST
sw[3]	led[0]	9.251	SLOW	2.867	FAST
sw[3]	led[1]	8.653	SLOW	2.649	FAST
sw[3]	led[7]	9.304	SLOW	2.933	FAST

The timing reports for encoders changed a little but utilization reports are the same

MUX1

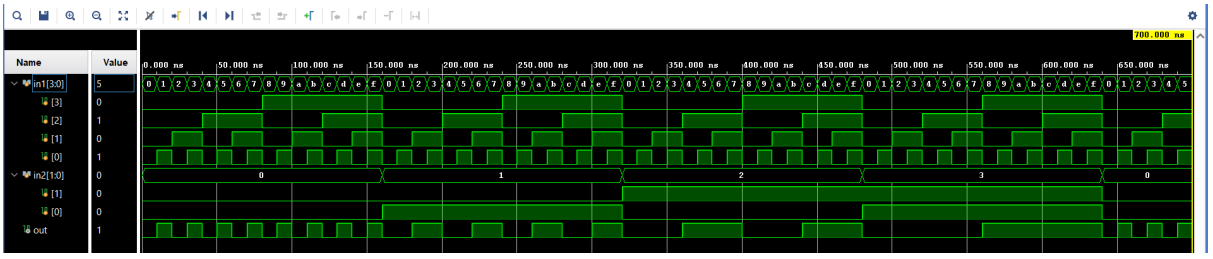
The code and testbench for Multiplexer:

```
module MUX(
input [3:0] D,
input [1:0] S,
output  O);

assign O = ((~S[1] & ~S[0] & D[0]) | (~S[1] & S[0] & D[1]) | (~S[0] & S[1] & D[2]) | (S[1] & S[0] & D[3]));

endmodule
```

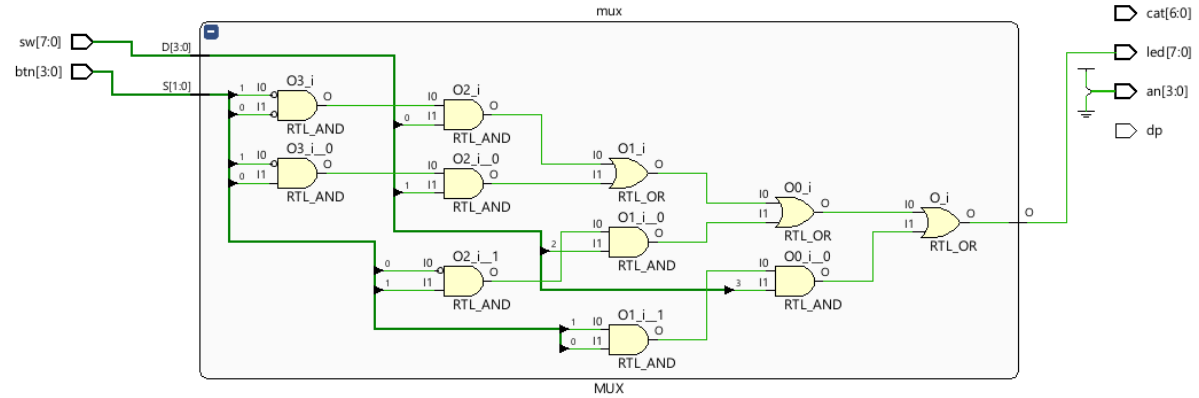
```
`timescale 1ns / 1ps

module MUX_tb;
reg [3:0] in1;
reg [1:0] in2;
wire out;

Top_Module dut(.sw(in1), .btn(in2), .led(out));

always
begin
#10 in1[0]=~in1[0];
end

always
begin
#20 in1[1]=~in1[1];
end

always
begin
#40 in1[2]=~in1[2];
end

always
begin
#80 in1[3]=~in1[3];
end

always
begin
#160 in2[0]=~in2[0];
end

always
begin
#320 in2[1]=~in2[1];
end

initial
begin
```

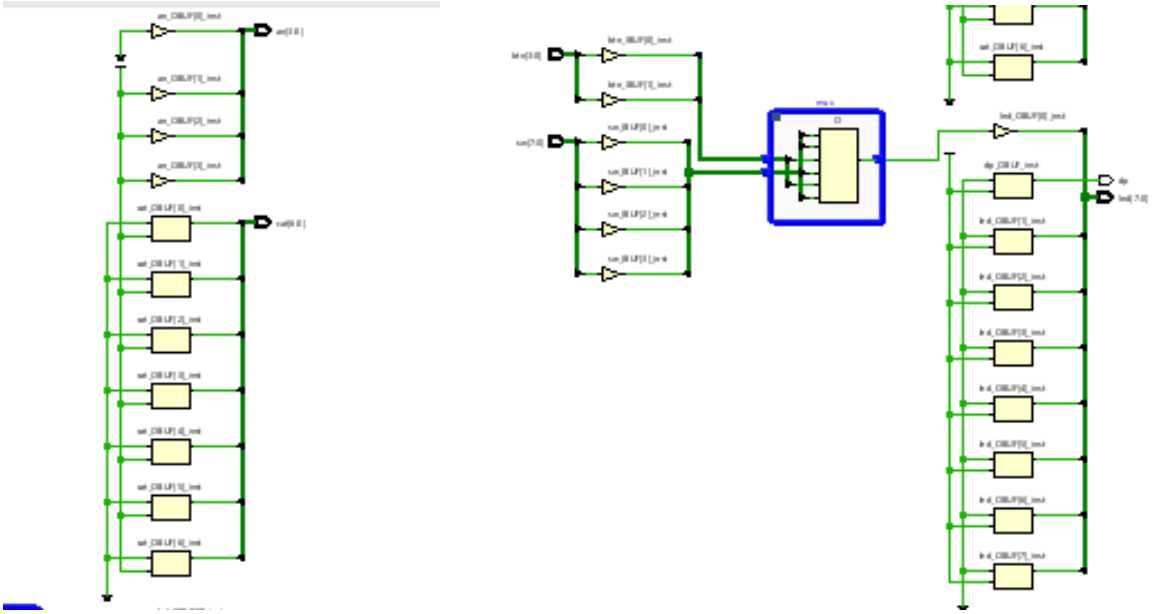
```
in1=0;
in2=0;
#700
$finish;
end

endmodule
```

Simulation wave:
all of the possible input combinations are obtained, and the output came as expected.



Rtl and Technology Schema for Mux1

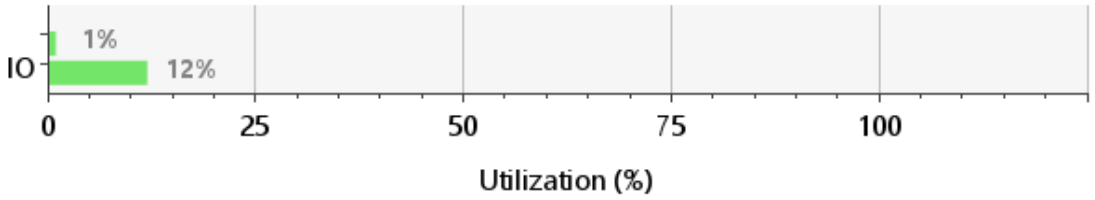




Timing and Utilization Summary before implementation for Mux1

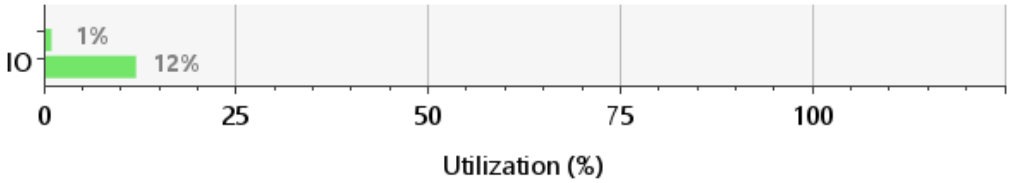
From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
btn[0]	led[0]	6.718	SLOW	2.183	FAST
btn[1]	led[0]	6.727	SLOW	2.191	FAST
sw[0]	led[0]	6.738	SLOW	2.203	FAST
sw[1]	led[0]	6.732	SLOW	2.196	FAST
sw[2]	led[0]	6.718	SLOW	2.182	FAST
sw[3]	led[0]	6.716	SLOW	2.181	FAST

Resource	Utilization	Available	Utilization %
LUT	1	32600	0.00
IO	26	210	12.38



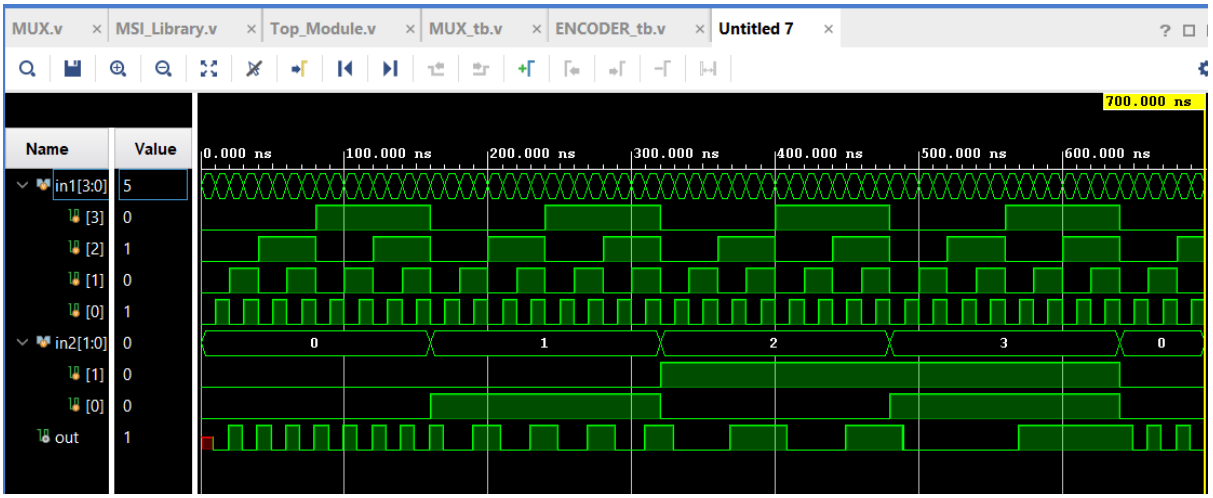
Timing and Utilization Summary after implementation for MUX1

Resource	Utilization	Available	Utilization %
LUT	1	32600	0.00
IO	26	210	12.38



From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
btn[0]	led[0]	9.164	SLOW	2.784	FAST
btn[1]	led[0]	8.408	SLOW	2.510	FAST
sw[0]	led[0]	8.997	SLOW	2.748	FAST
sw[1]	led[0]	8.953	SLOW	2.687	FAST
sw[2]	led[0]	8.623	SLOW	2.602	FAST
sw[3]	led[0]	9.237	SLOW	2.848	FAST

simulation wave after implementation

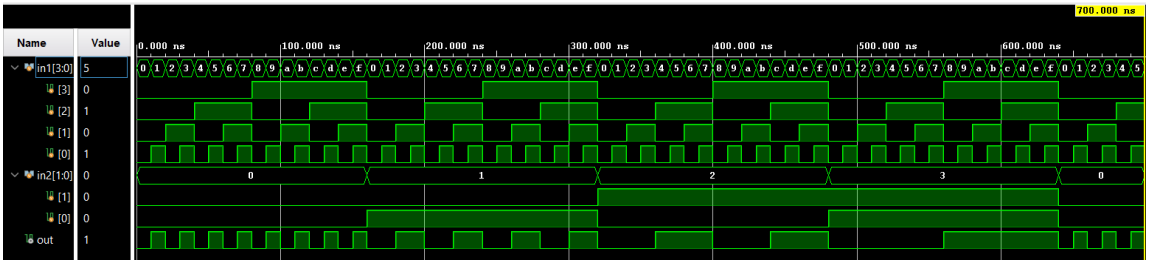


MUX2

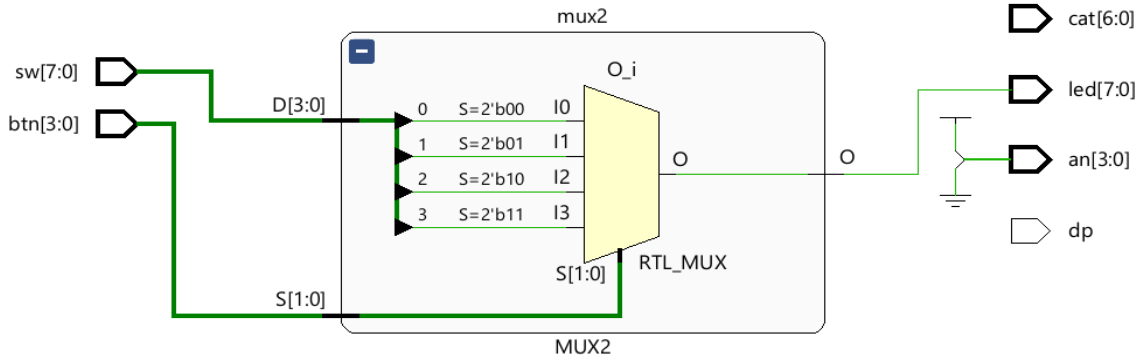
Code for Multiplexer with case structure:

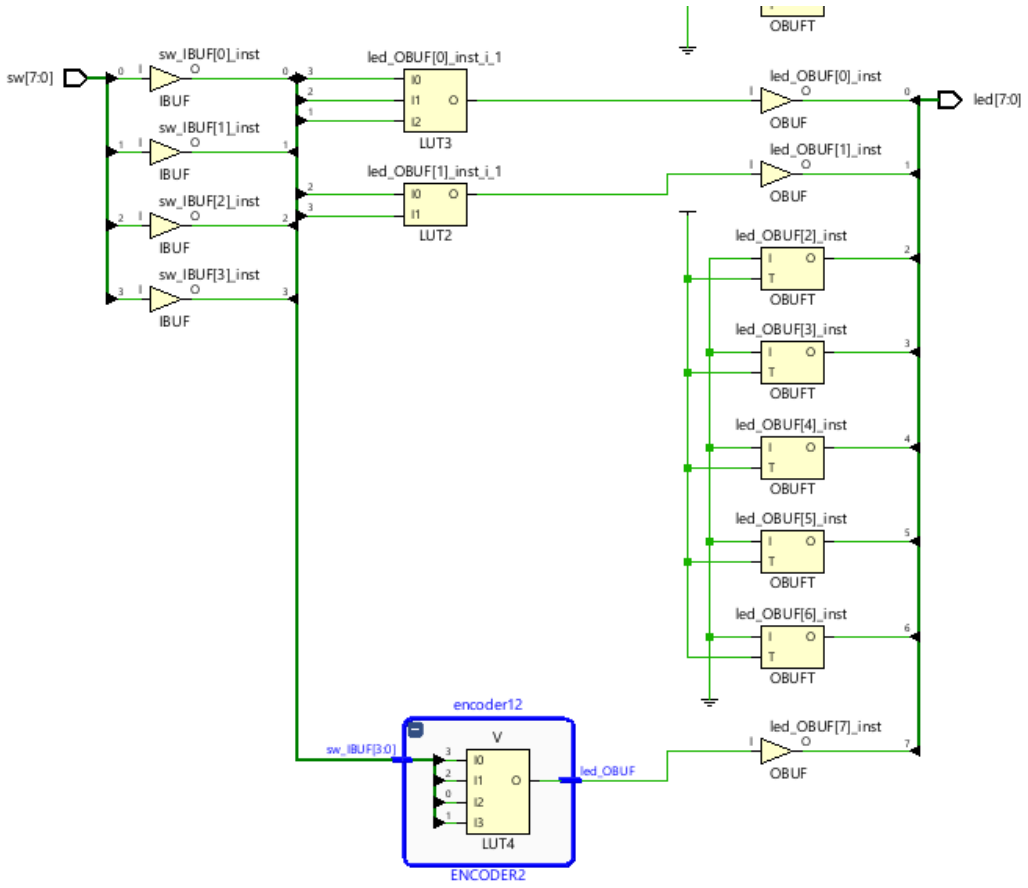
```
module MUX2(
input [3:0] D,
input [1:0] S,
output reg O
    );

    always@(S, D) begin
   case(S)
    2'b00   :  O=D[0];
    2'b01   :  O=D[1];
    2'b10   :  O=D[2];
    2'b11   :  O=D[3];
    default : O=1'bZ;
    endcase
    end
endmodule
```

Simulation wave:
all of the possible input combinations are obtained, and the output came as expected.



RTL and Technology Schematic for Mux2:



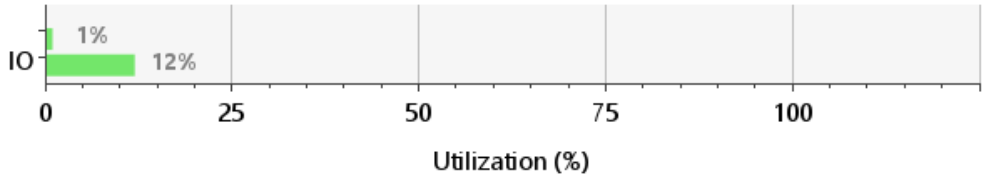


Utilization Summary and Time Report for Mux2:

Same with MUX1.

Summary

Resource	Utilization	Available	Utilization %
LUT	1	32600	0.00
IO	26	210	12.38



Combinational Delays

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
btn[0]	led[0]	9.164	SLOW	2.784	FAST
btn[1]	led[0]	8.408	SLOW	2.510	FAST
sw[0]	led[0]	8.997	SLOW	2.748	FAST
sw[1]	led[0]	8.953	SLOW	2.687	FAST
sw[2]	led[0]	8.623	SLOW	2.602	FAST
sw[3]	led[0]	9.237	SLOW	2.848	FAST

The utilzatşon and timing reports for the multiplexers did not change.

demux

Code and Testbench for Demultiplexer:

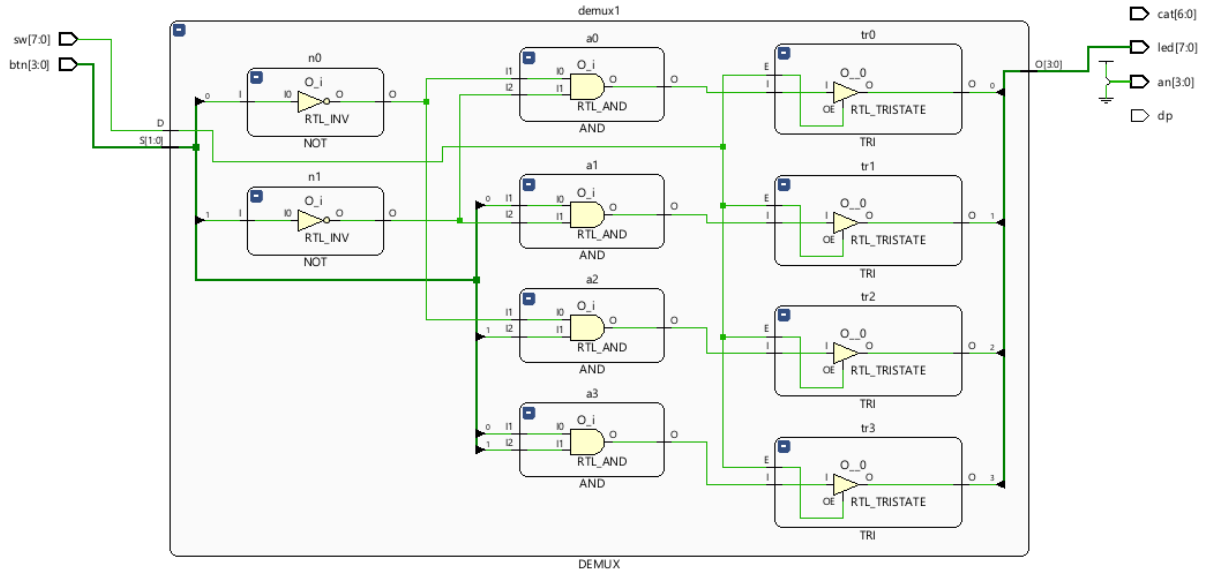
```
module DEMUX(
input D,
input [1:0]S,
output [3:0] O);

    wire [5:0]temp;

    NOT n0(S[0],temp[0]);
    NOT n1(S[1],temp[1]);
```

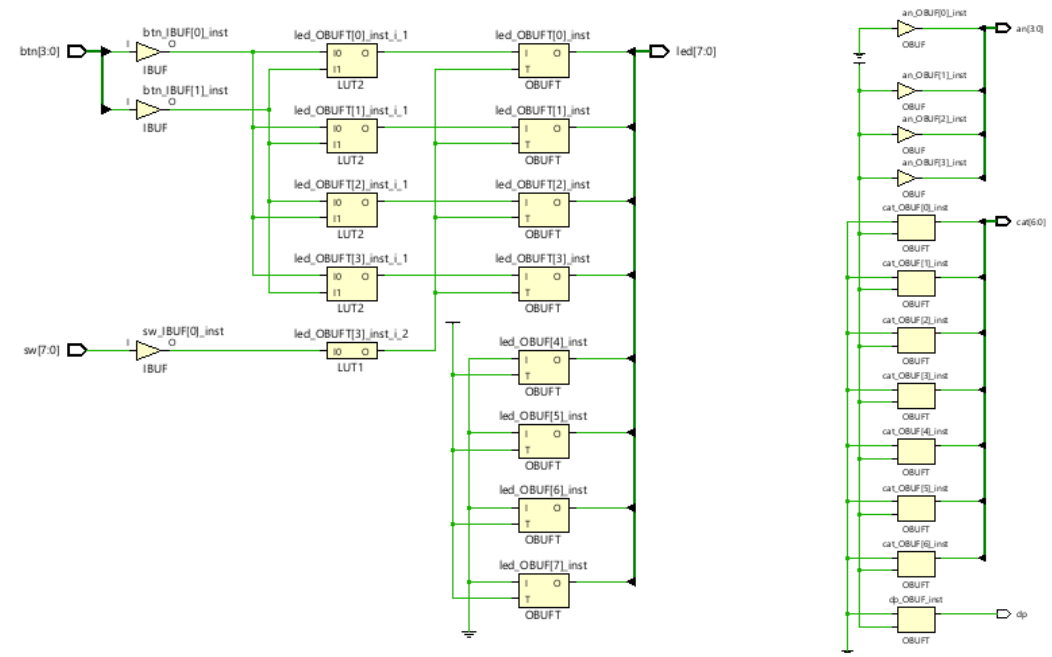
```
    AND a0(temp[0],temp[1],temp[2]);
    AND a1(S[0],temp[1],temp[3]);
    AND a2(temp[0],S[1],temp[4]);
    AND a3(S[0],S[1],temp[5]);

    TRI tr0(temp[2],D,O[0]);
    TRI tr1(temp[3],D,O[1]);
    TRI tr2(temp[4],D,O[2]);
    TRI tr3(temp[5],D,O[3]);
endmodule
```

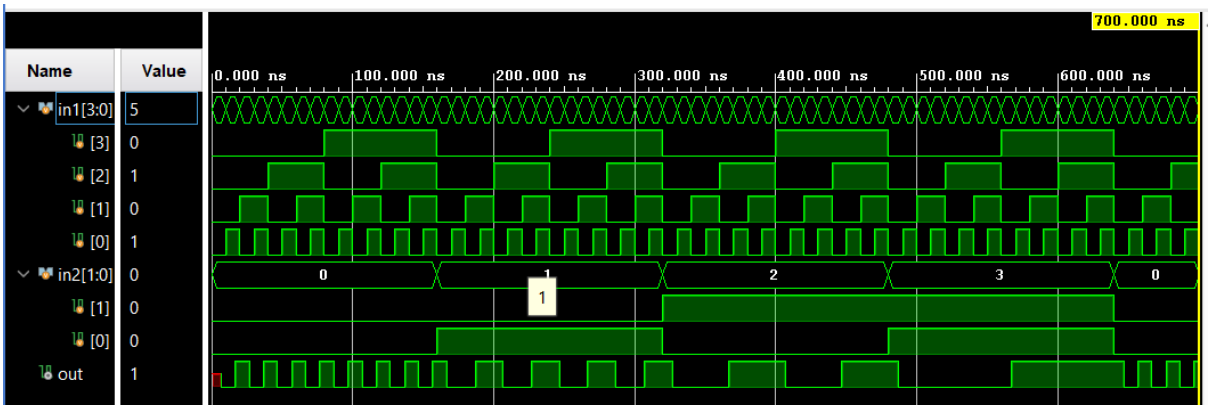
```
module DEMUX_tb;
reg in1;
reg [1:0] in2;
wire [3:0] out;

Top_Module dut(.sw(in1), .btn(in2), .led(out));

always
begin
#5 in2[0]=~in2[0];
end
always
begin
#10 in2[1]=~in2[1];
end
always
begin
#20 in1=~in1;
end
initial
begin
in1=0;
in2=0;
#50
$finish;
end

endmodule
```

Simulation wave:
all of the possible input combinations are obtained, and the output came as expected.

RTL and Technology Schematic



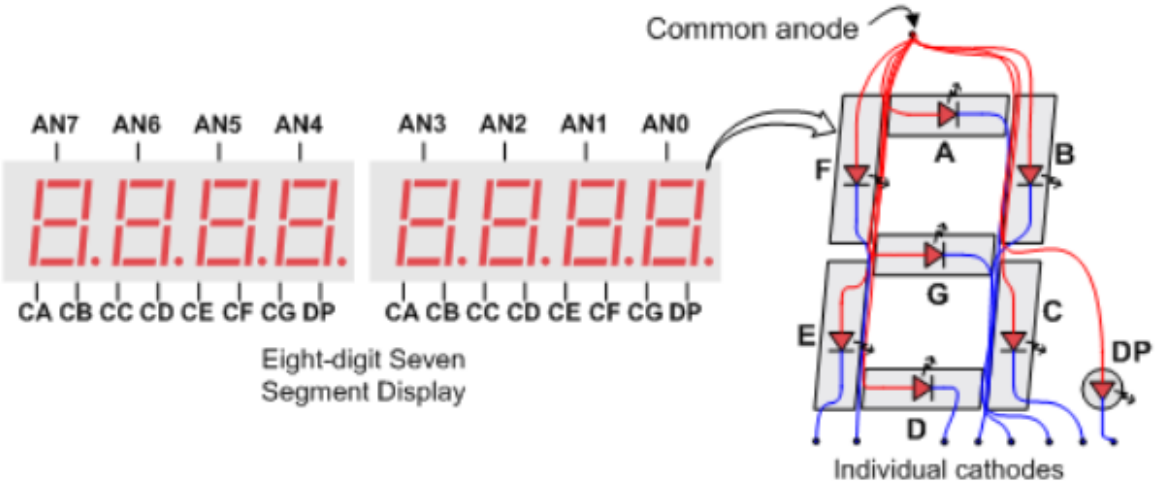




7 Segment Display

The seven-segment LEDs on the Nexys A7 FPGA development board consist of seven individual LED segments arranged in a pattern that allows the display of numbers and some letters. These segments can be controlled independently. The display can be of either "common anode" or "common cathode" type. In a common anode display, the positive terminals of all segments are connected together, and the negative terminals are controlled individually. In a common cathode display, the negative terminals of all segments are connected, and the positive terminals are controlled individually.

These displays are commonly used to represent numbers and certain characters. Each segment can be illuminated to represent a specific numeral or letter. In FPGA-based systems, digital I/O pins of the FPGA are used to control the seven-segment displays. The FPGA design includes logic circuits to determine the appropriate combination to display a specific character by selectively turning on and off the segments. Multiplexing techniques may also be employed when multiple seven-segment displays are used, where the displays are rapidly switched, giving the illusion that all displays are active simultaneously.


AN7 AN6 AN5 AN4
AN3 AN2 AN1 AN0
CA CB CC CD CE CF CG DP
CA CB CC CD CE CF CG DP
Eight-digit Seven Segment Display
Common anode
A
B
C
D
E
F
G
DP
Individual cathodes