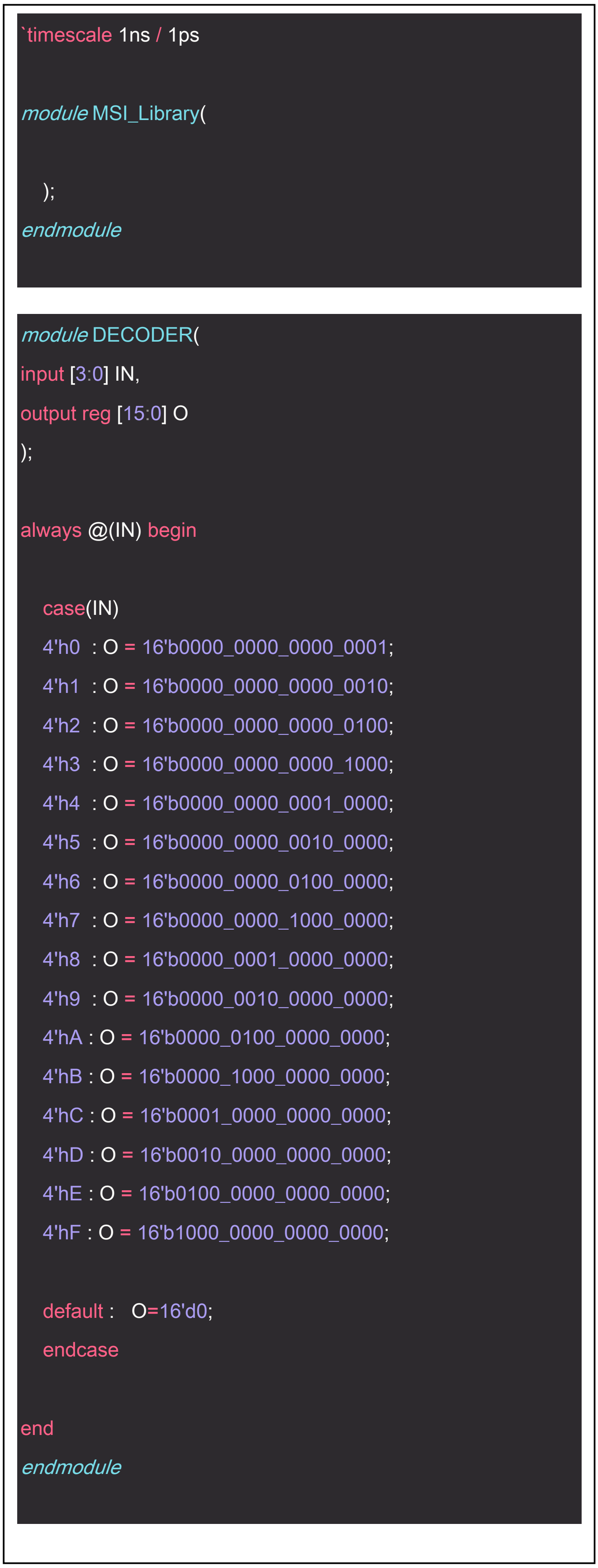
DIGITAL SYSTEM DESIGN APPLICATION PROJECT 2

İBRAHIM SAMED YALÇIN

040180055



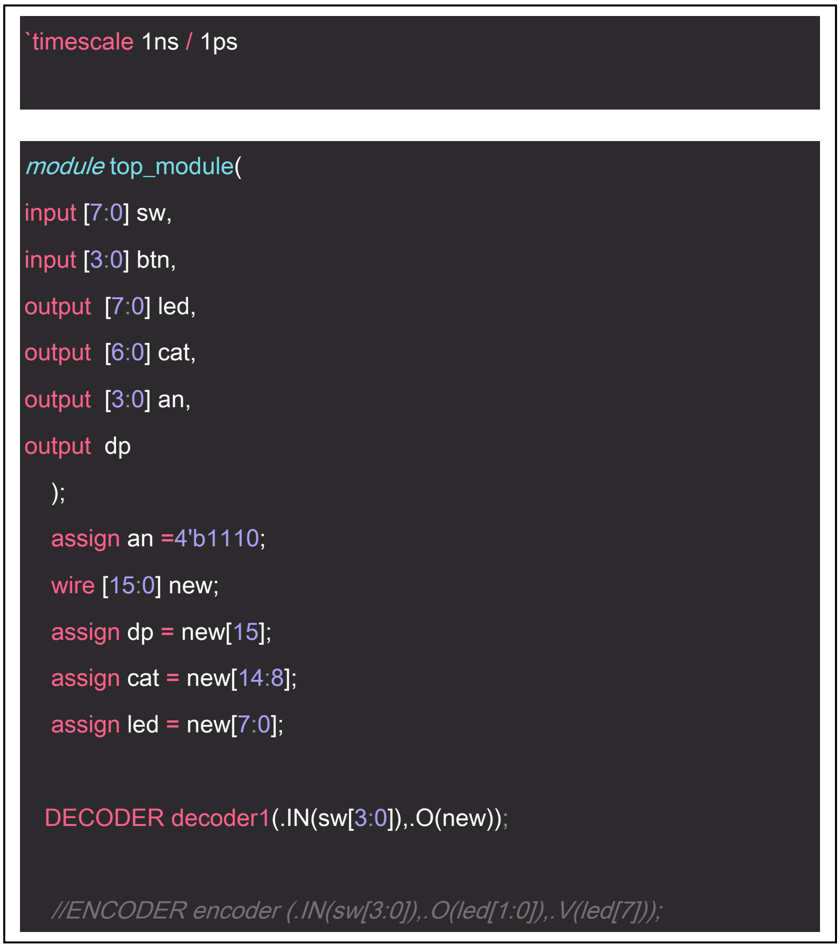
```
`timescale 1ns / 1ps

module MSI_Library(

    );
endmodule
```

```
module DECODER(
input [3:0] IN,
output reg [15:0] O
);

always @(IN) begin

    case(IN)
    4'h0  : O = 16'b0000_0000_0000_0001;
    4'h1  : O = 16'b0000_0000_0000_0010;
    4'h2  : O = 16'b0000_0000_0000_0100;
    4'h3  : O = 16'b0000_0000_0000_1000;
    4'h4  : O = 16'b0000_0000_0001_0000;
    4'h5  : O = 16'b0000_0000_0010_0000;
    4'h6  : O = 16'b0000_0000_0100_0000;
    4'h7  : O = 16'b0000_0000_1000_0000;
    4'h8  : O = 16'b0000_0001_0000_0000;
    4'h9  : O = 16'b0000_0010_0000_0000;
    4'hA : O = 16'b0000_0100_0000_0000;
    4'hB : O = 16'b0000_1000_0000_0000;
    4'hC : O = 16'b0001_0000_0000_0000;
    4'hD : O = 16'b0010_0000_0000_0000;
    4'hE : O = 16'b0100_0000_0000_0000;
    4'hF : O = 16'b1000_0000_0000_0000;

    default :   O=16'd0;
    endcase

end
endmodule
```

DECODER

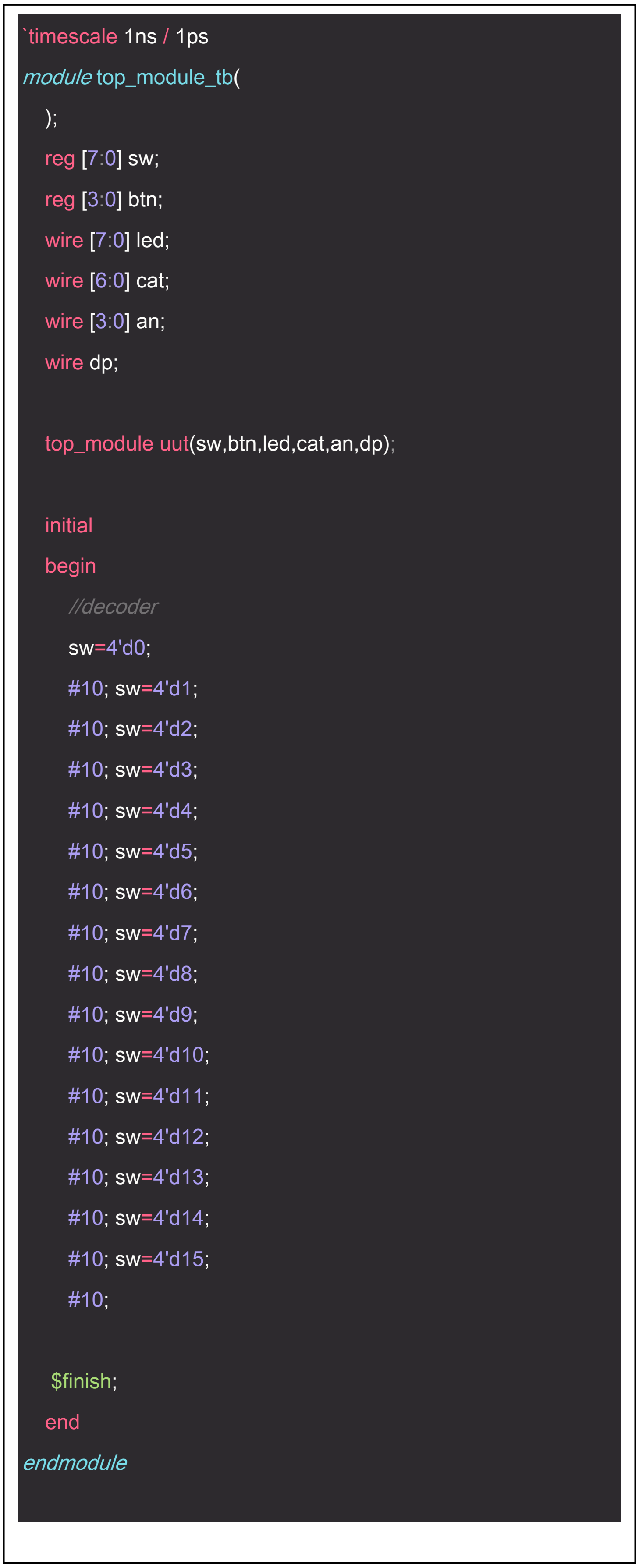
MSI LİBRARY

TOP MODULE



```
`timescale 1ns / 1ps

module top_module(
input [7:0] sw,
input [3:0] btn,
output  [7:0] led,
output  [6:0] cat,
output  [3:0] an,
output  dp
    );
    assign an =4'b1110;
    wire [15:0] new;
    assign dp = new[15];
    assign cat = new[14:8];
    assign led = new[7:0];

   DECODER decoder1(.IN(sw[3:0]),.O(new));

    //ENCODER encoder (.IN(sw[3:0]),.O(led[1:0]),.V(led[7]));
```

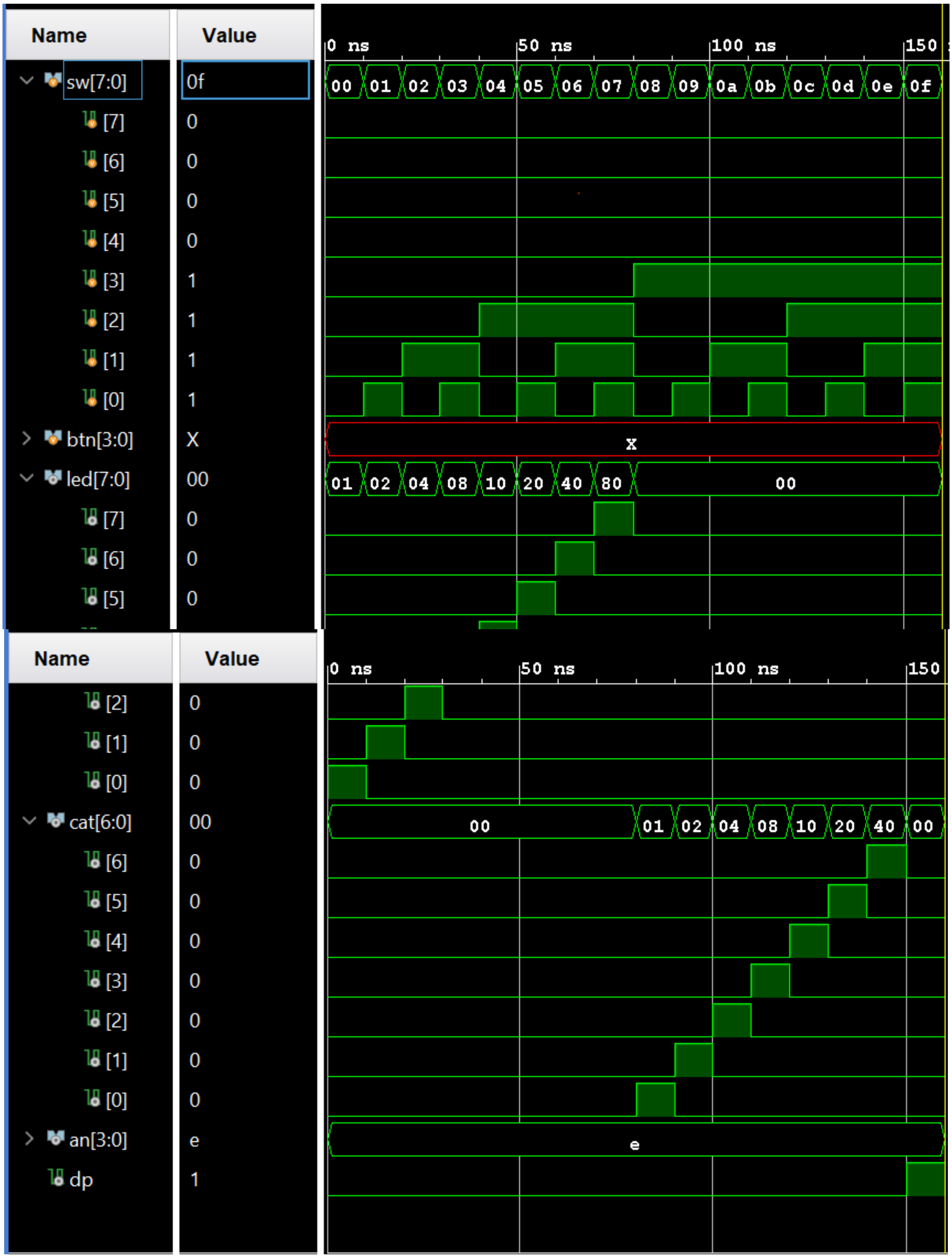
TEST BENCH



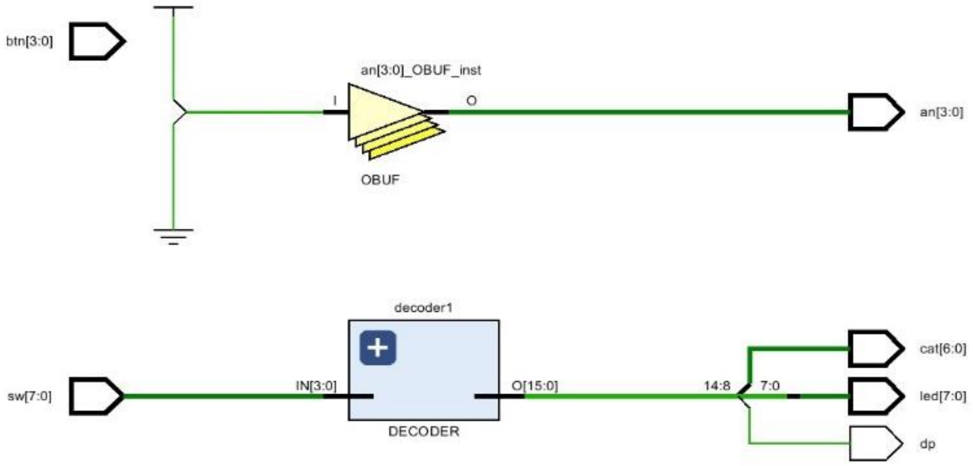
```
`timescale 1ns / 1ps
module top_module_tb(
    );
    reg [7:0] sw;
    reg [3:0] btn;
    wire [7:0] led;
    wire [6:0] cat;
    wire [3:0] an;
    wire dp;

    top_module uut(sw,btn,led,cat,an,dp);

    initial
    begin
        //decoder
        sw=4'd0;
        #10; sw=4'd1;
        #10; sw=4'd2;
        #10; sw=4'd3;
        #10; sw=4'd4;
        #10; sw=4'd5;
        #10; sw=4'd6;
        #10; sw=4'd7;
        #10; sw=4'd8;
        #10; sw=4'd9;
        #10; sw=4'd10;
        #10; sw=4'd11;
        #10; sw=4'd12;
        #10; sw=4'd13;
        #10; sw=4'd14;
        #10; sw=4'd15;
        #10;

     $finish;
    end
endmodule
```

BEHAVİORAL

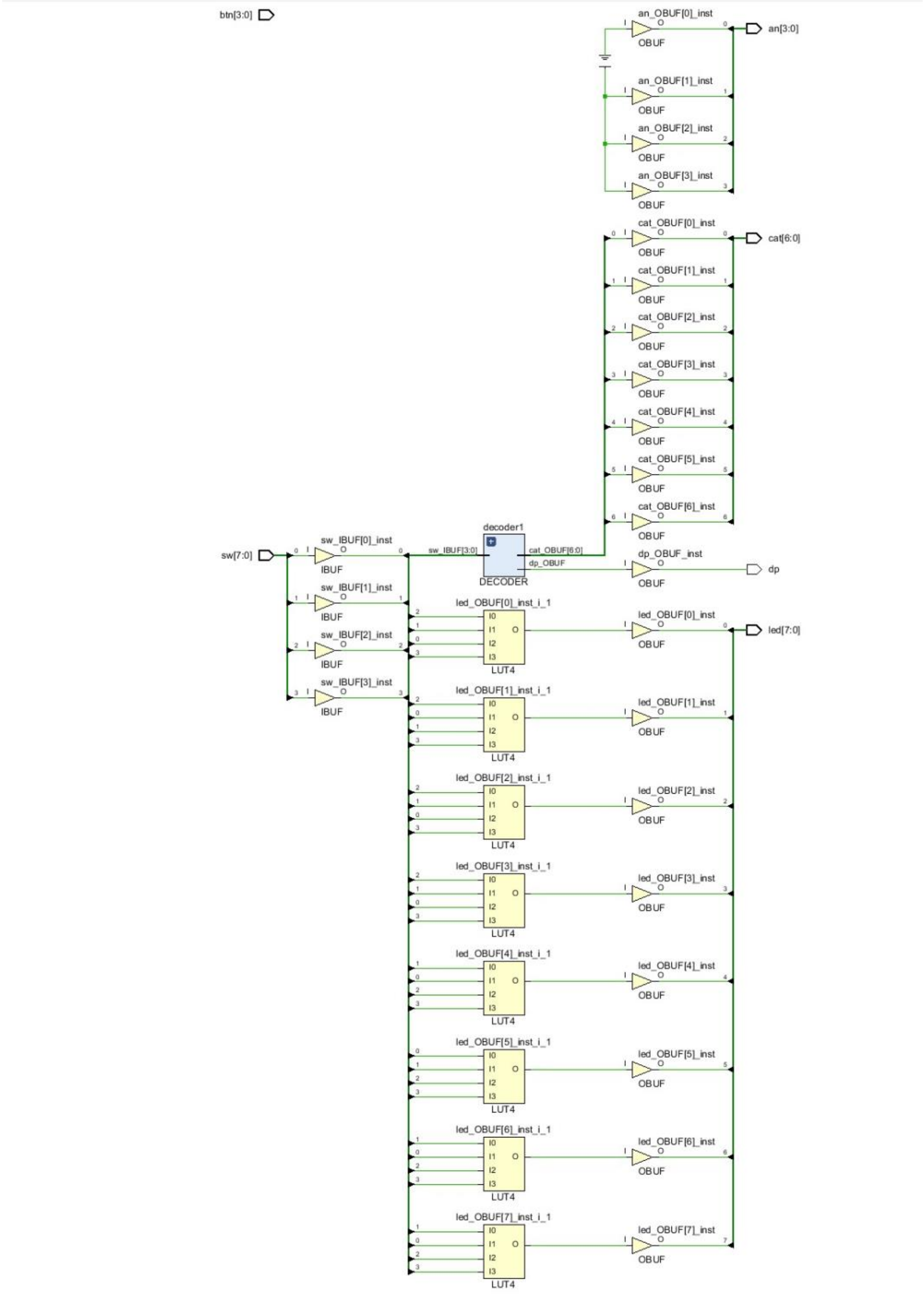
SHEMATİC


Name
Value
0 ns
50 ns
100 ns
150
sw[7:0]
0f
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
[7] 0
[6] 0
[5] 0
[4] 0
[3] 1
[2] 1
[1] 1
[0] 1
btn[3:0]
X
x
led[7:0]
00
01 02 04 08 10 20 40 80 00
[7] 0
[6] 0
[5] 0
Name
Value
0 ns
50 ns
100 ns
150
[2] 0
[1] 0
[0] 0
cat[6:0]
00
00 01 02 04 08 10 20 40 00
[6] 0
[5] 0
[4] 0
[3] 0
[2] 0
[1] 0
[0] 0
an[3:0]
e
e
dp
1

RTL SHEMA

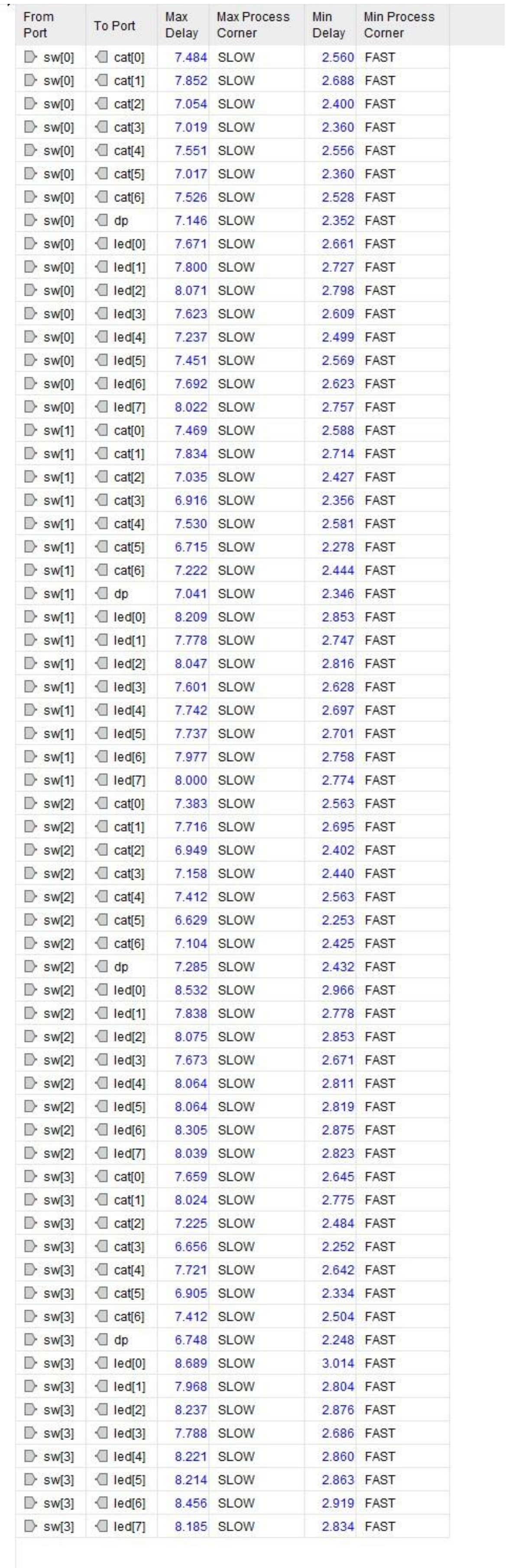


TECHNOLOGIC SHEMA



THERE ARE 8 LUTS

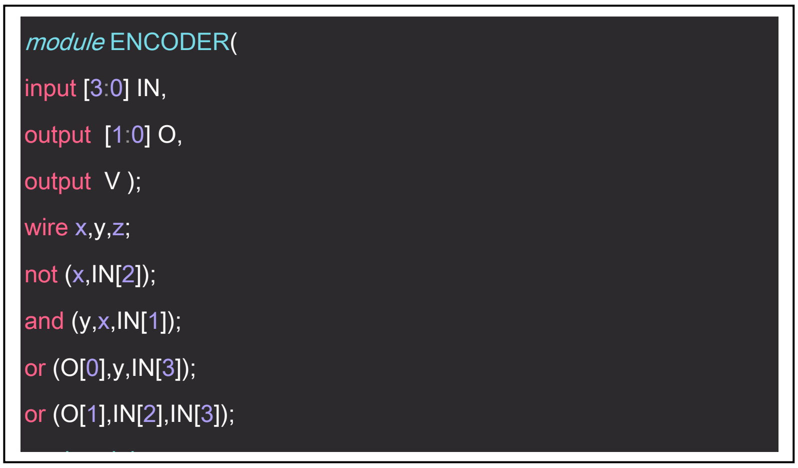
DELAY:



From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
sw[0]	cat[0]	7.484	SLOW	2.560	FAST
sw[0]	cat[1]	7.852	SLOW	2.688	FAST
sw[0]	cat[2]	7.054	SLOW	2.400	FAST
sw[0]	cat[3]	7.019	SLOW	2.360	FAST
sw[0]	cat[4]	7.551	SLOW	2.556	FAST
sw[0]	cat[5]	7.017	SLOW	2.360	FAST
sw[0]	cat[6]	7.526	SLOW	2.528	FAST
sw[0]	dp	7.146	SLOW	2.352	FAST
sw[0]	led[0]	7.671	SLOW	2.661	FAST
sw[0]	led[1]	7.800	SLOW	2.727	FAST
sw[0]	led[2]	8.071	SLOW	2.798	FAST
sw[0]	led[3]	7.623	SLOW	2.609	FAST
sw[0]	led[4]	7.237	SLOW	2.499	FAST
sw[0]	led[5]	7.451	SLOW	2.569	FAST
sw[0]	led[6]	7.692	SLOW	2.623	FAST
sw[0]	led[7]	8.022	SLOW	2.757	FAST
sw[1]	cat[0]	7.469	SLOW	2.588	FAST
sw[1]	cat[1]	7.834	SLOW	2.714	FAST
sw[1]	cat[2]	7.035	SLOW	2.427	FAST
sw[1]	cat[3]	6.916	SLOW	2.356	FAST
sw[1]	cat[4]	7.530	SLOW	2.581	FAST
sw[1]	cat[5]	6.715	SLOW	2.278	FAST
sw[1]	cat[6]	7.222	SLOW	2.444	FAST
sw[1]	dp	7.041	SLOW	2.346	FAST
sw[1]	led[0]	8.209	SLOW	2.853	FAST
sw[1]	led[1]	7.778	SLOW	2.747	FAST
sw[1]	led[2]	8.047	SLOW	2.816	FAST
sw[1]	led[3]	7.601	SLOW	2.628	FAST
sw[1]	led[4]	7.742	SLOW	2.697	FAST
sw[1]	led[5]	7.737	SLOW	2.701	FAST
sw[1]	led[6]	7.977	SLOW	2.758	FAST
sw[1]	led[7]	8.000	SLOW	2.774	FAST
sw[2]	cat[0]	7.383	SLOW	2.563	FAST
sw[2]	cat[1]	7.716	SLOW	2.695	FAST
sw[2]	cat[2]	6.949	SLOW	2.402	FAST
sw[2]	cat[3]	7.158	SLOW	2.440	FAST
sw[2]	cat[4]	7.412	SLOW	2.563	FAST
sw[2]	cat[5]	6.629	SLOW	2.253	FAST
sw[2]	cat[6]	7.104	SLOW	2.425	FAST
sw[2]	dp	7.285	SLOW	2.432	FAST
sw[2]	led[0]	8.532	SLOW	2.966	FAST
sw[2]	led[1]	7.838	SLOW	2.778	FAST
sw[2]	led[2]	8.075	SLOW	2.853	FAST
sw[2]	led[3]	7.673	SLOW	2.671	FAST
sw[2]	led[4]	8.064	SLOW	2.811	FAST
sw[2]	led[5]	8.064	SLOW	2.819	FAST
sw[2]	led[6]	8.305	SLOW	2.875	FAST
sw[2]	led[7]	8.039	SLOW	2.823	FAST
sw[3]	cat[0]	7.659	SLOW	2.645	FAST
sw[3]	cat[1]	8.024	SLOW	2.775	FAST
sw[3]	cat[2]	7.225	SLOW	2.484	FAST
sw[3]	cat[3]	6.656	SLOW	2.252	FAST
sw[3]	cat[4]	7.721	SLOW	2.642	FAST
sw[3]	cat[5]	6.905	SLOW	2.334	FAST
sw[3]	cat[6]	7.412	SLOW	2.504	FAST
sw[3]	dp	6.748	SLOW	2.248	FAST
sw[3]	led[0]	8.689	SLOW	3.014	FAST
sw[3]	led[1]	7.968	SLOW	2.804	FAST
sw[3]	led[2]	8.237	SLOW	2.876	FAST
sw[3]	led[3]	7.788	SLOW	2.686	FAST
sw[3]	led[4]	8.221	SLOW	2.860	FAST
sw[3]	led[5]	8.214	SLOW	2.863	FAST
sw[3]	led[6]	8.456	SLOW	2.919	FAST
sw[3]	led[7]	8.185	SLOW	2.834	FAST

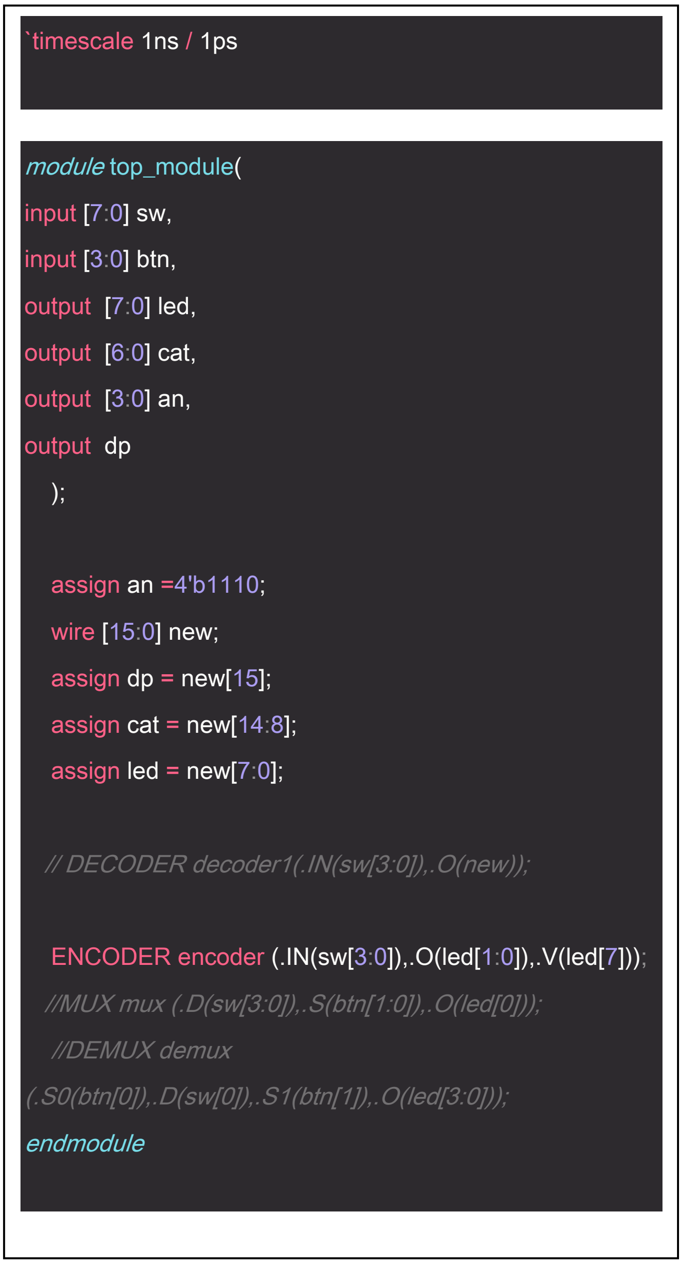
ENCODER

MSI LİBRARY



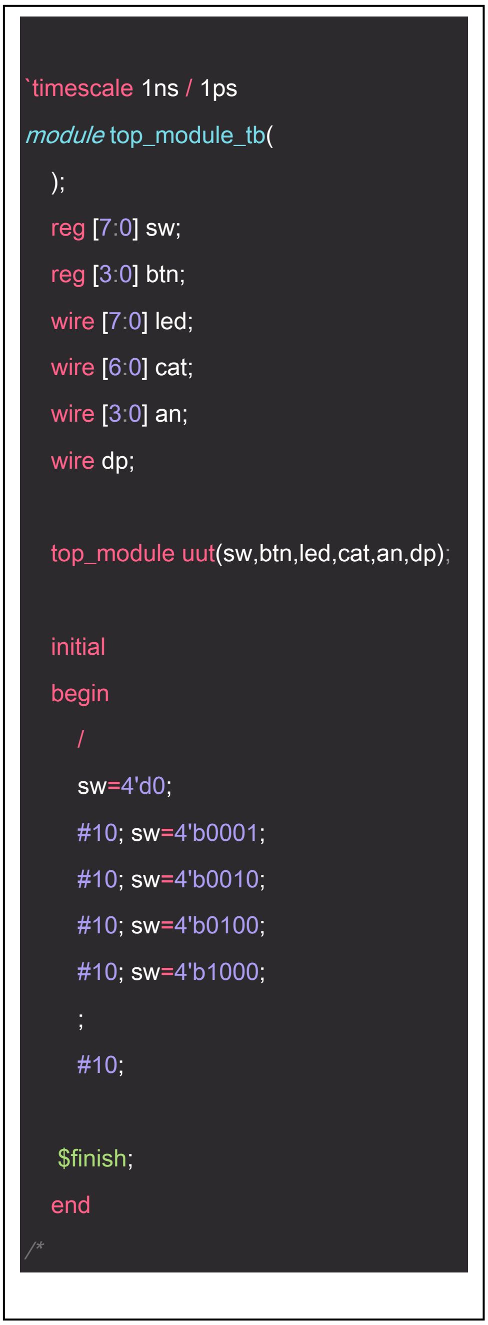
```
module ENCODER(
input [3:0] IN,
output  [1:0] O,
output  V );
wire x,y,z;
not (x,IN[2]);
and (y,x,IN[1]);
or (O[0],y,IN[3]);
or (O[1],IN[2],IN[3]);
```

TOP MODULE



```
`timescale 1ns / 1ps

module top_module(
input [7:0] sw,
input [3:0] btn,
output  [7:0] led,
output  [6:0] cat,
output  [3:0] an,
output  dp
    );

    assign an =4'b1110;
    wire [15:0] new;
    assign dp = new[15];
    assign cat = new[14:8];
    assign led = new[7:0];

   // DECODER decoder1(.IN(sw[3:0]),.O(new));

    ENCODER encoder (.IN(sw[3:0]),.O(led[1:0]),.V(led[7]));
   //MUX mux (.D(sw[3:0]),.S(btn[1:0]),.O(led[0]));
    //DEMUX demux
(.S0(btn[0]),.D(sw[0]),.S1(btn[1]),.O(led[3:0]));
endmodule
```

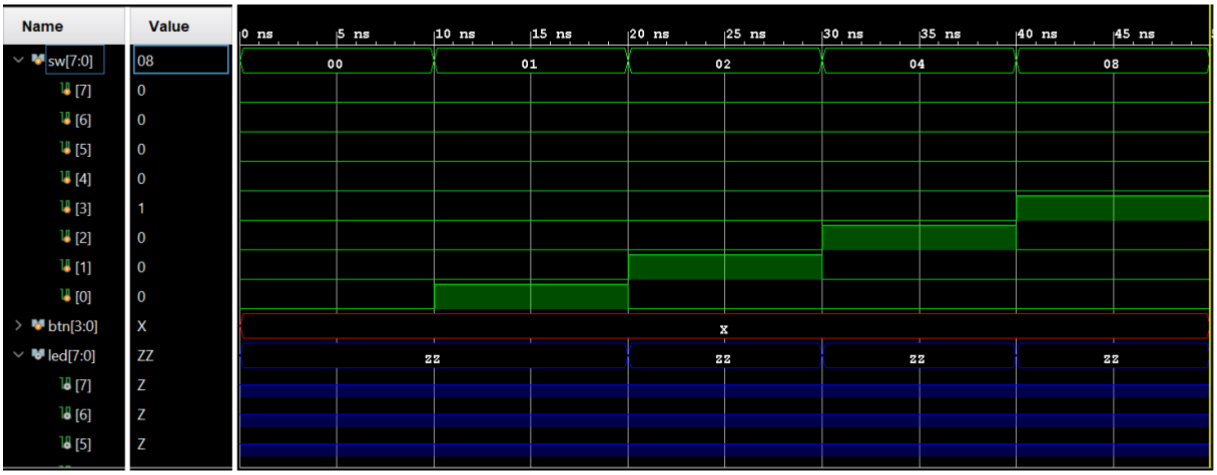
TESTSTBENCH



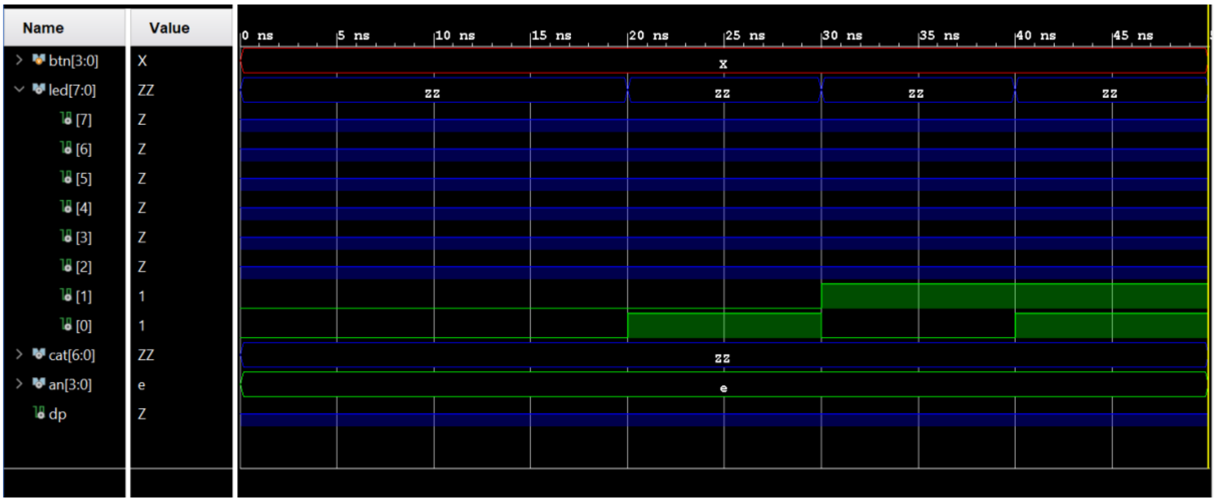
```
`timescale 1ns / 1ps
module top_module_tb(
    );
    reg [7:0] sw;
    reg [3:0] btn;
    wire [7:0] led;
    wire [6:0] cat;
    wire [3:0] an;
    wire dp;

    top_module uut(sw,btn,led,cat,an,dp);

    initial
    begin
        /
        sw=4'd0;
        #10; sw=4'b0001;
        #10; sw=4'b0010;
        #10; sw=4'b0100;
        #10; sw=4'b1000;
        ;
        #10;

     $finish;
    end
/*
```

BEHAVIORAL

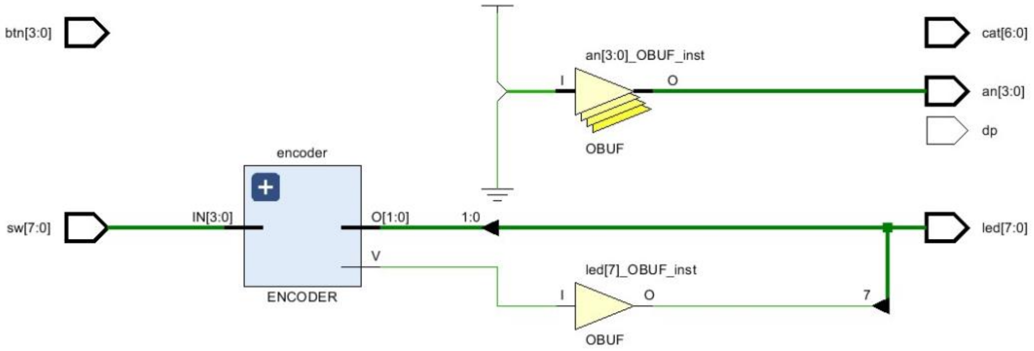
SHEMA



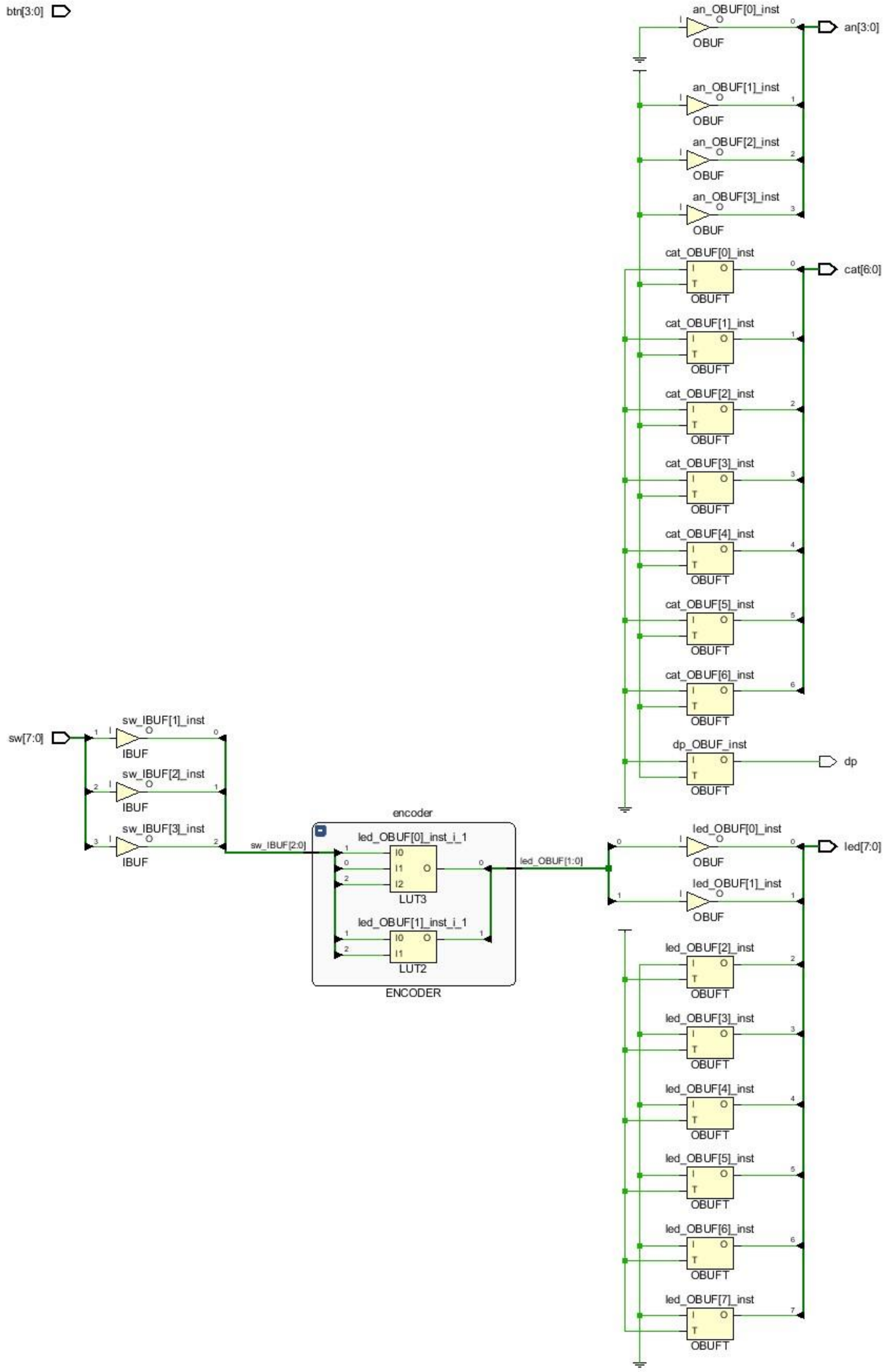
Name	Value
sw[7:0]	08
[7]	0
[6]	0
[5]	0
[4]	0
[3]	1
[2]	0
[1]	0
[0]	0
btn[3:0]	X
led[7:0]	ZZ
[7]	Z
[6]	Z
[5]	Z


Name
Value
btn[3:0]
X
led[7:0]
ZZ
[7]
Z
[6]
Z
[5]
Z
[4]
Z
[3]
Z
[2]
Z
[1]
1
[0]
1
cat[6:0]
ZZ
an[3:0]
e
dp
Z
0 ns
5 ns
10 ns
15 ns
20 ns
25 ns
30 ns
35 ns
40 ns
45 ns
X
ZZ
ZZ
ZZ
ZZ
ZZ
e

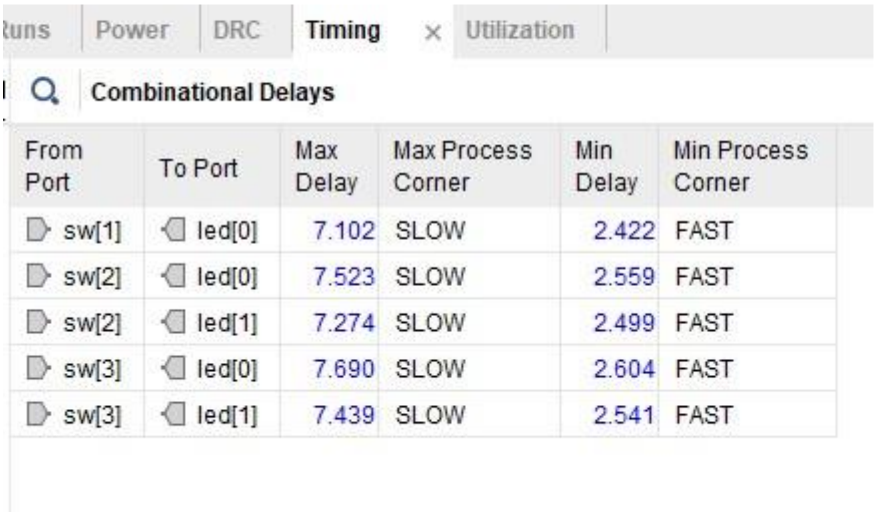
RTL SHEMA


btn[3:0]
an[3:0]_OBUF_inst
I
O
OBUF
cat[6:0]
an[3:0]
dp
encoder
sw[7:0]
IN[3:0]
O[1:0]
V
ENCODER
1:0
led[7:0]
led[7]_OBUF_inst
I
O
7
OBUF

TECHNOLOGIC SEHMA



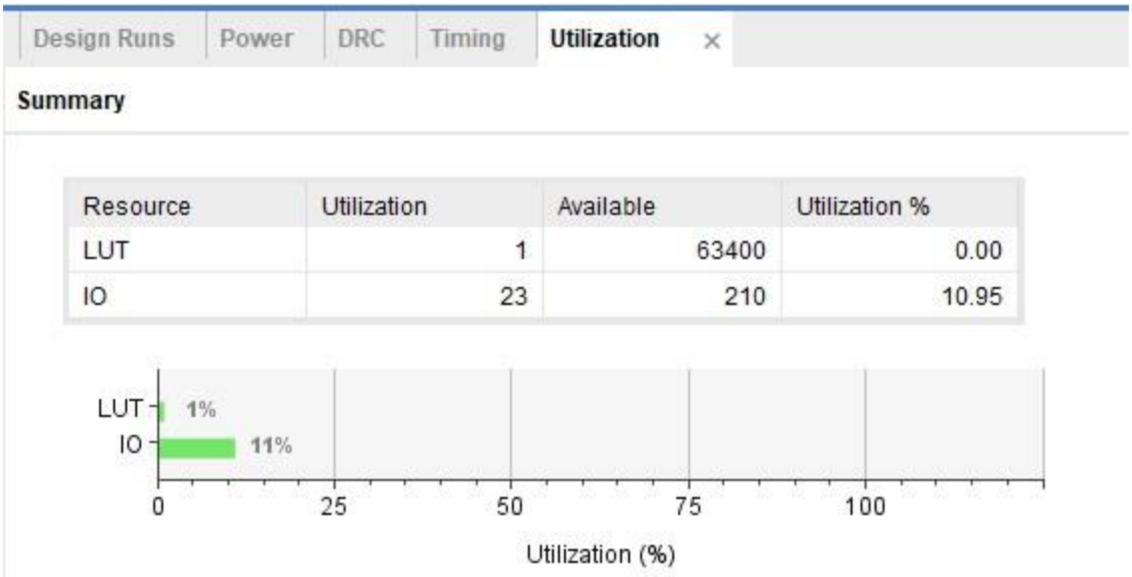
TİMİNG DELAY REPORT



Combinational Delays

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
sw[1]	led[0]	7.102	SLOW	2.422	FAST
sw[2]	led[0]	7.523	SLOW	2.559	FAST
sw[2]	led[1]	7.274	SLOW	2.499	FAST
sw[3]	led[0]	7.690	SLOW	2.604	FAST
sw[3]	led[1]	7.439	SLOW	2.541	FAST

UTILIZATION

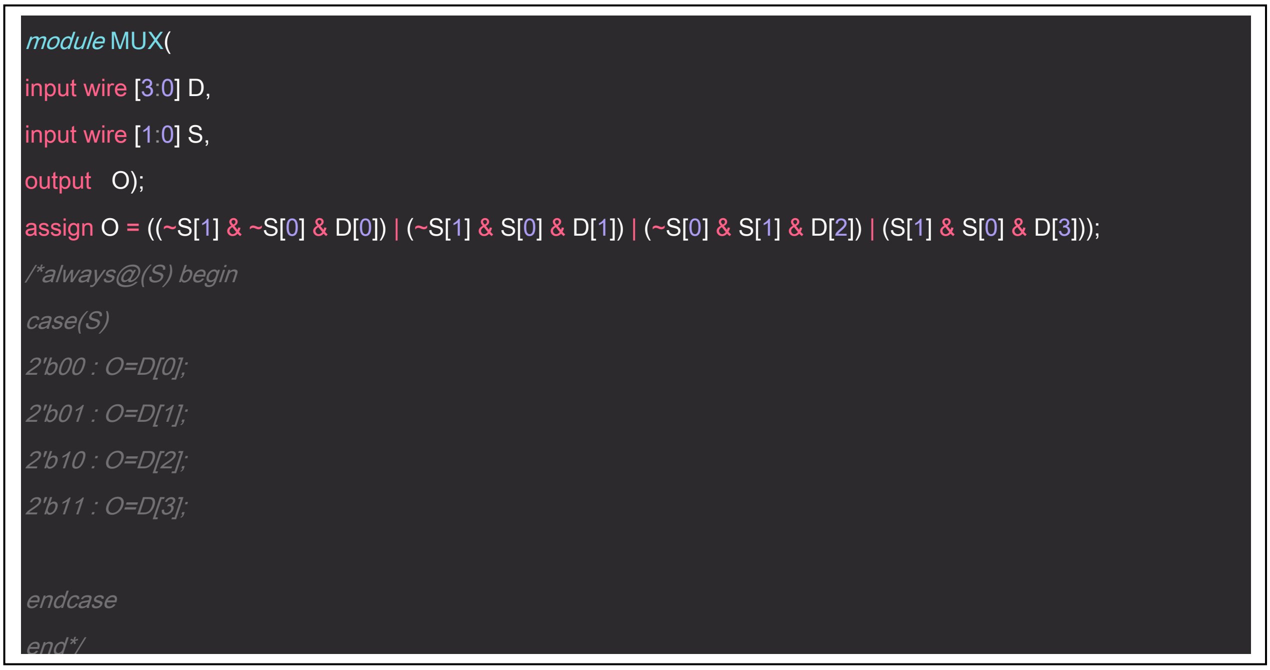


Summary

Resource	Utilization	Available	Utilization %
LUT	1	63400	0.00
IO	23	210	10.95

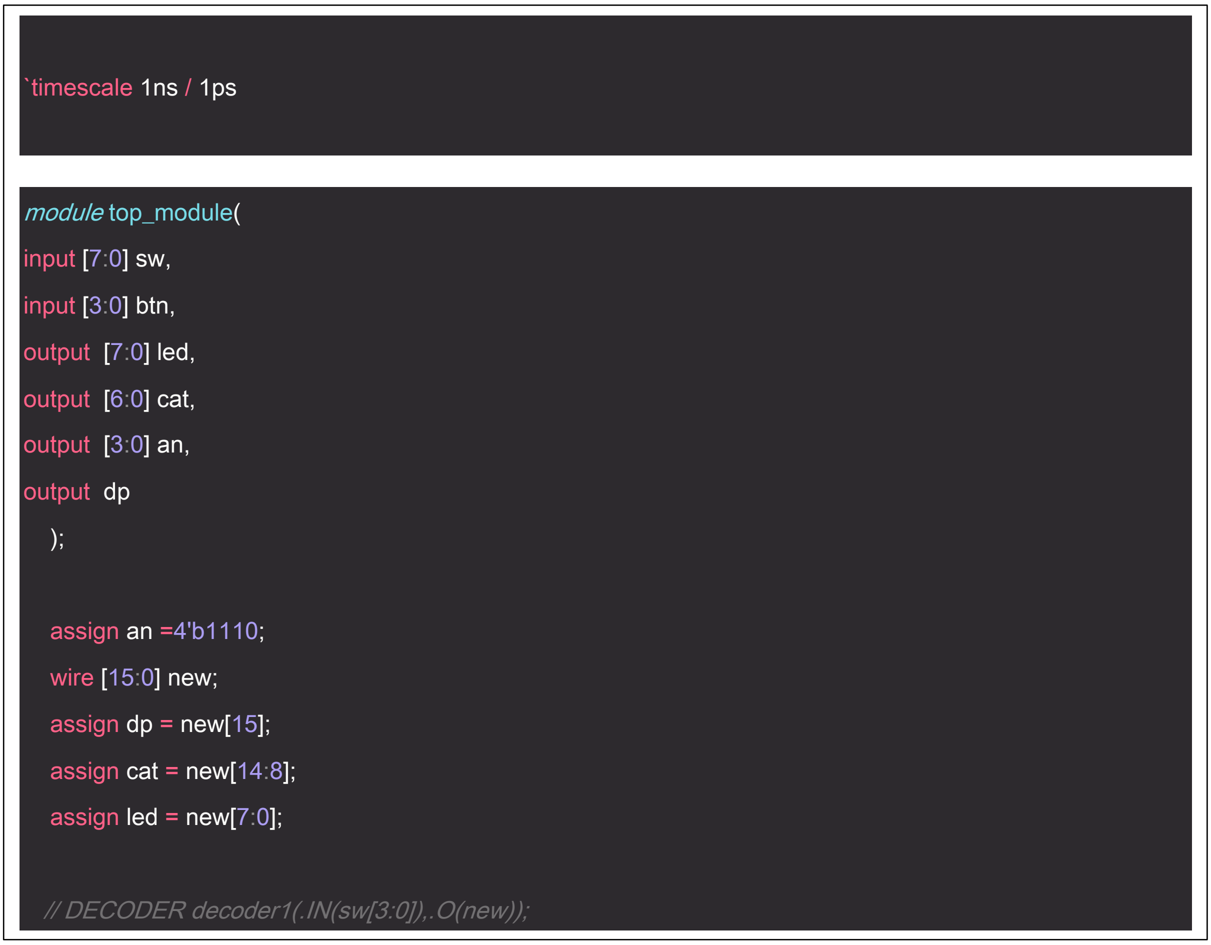
MULTIPLEXER

MSI LIBRARY



```
module MUX(
input wire [3:0] D,
input wire [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]));
/*always@(S) begin
case(S)
2'b00 : O=D[0];
2'b01 : O=D[1];
2'b10 : O=D[2];
2'b11 : O=D[3];

endcase
end*/
```

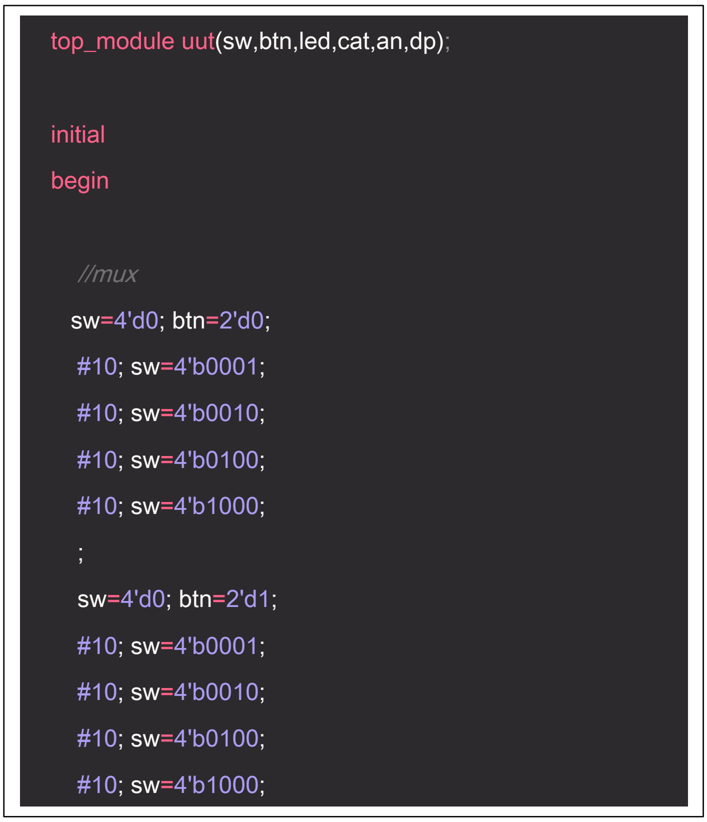
TOP MODULE



```
`timescale 1ns / 1ps
```

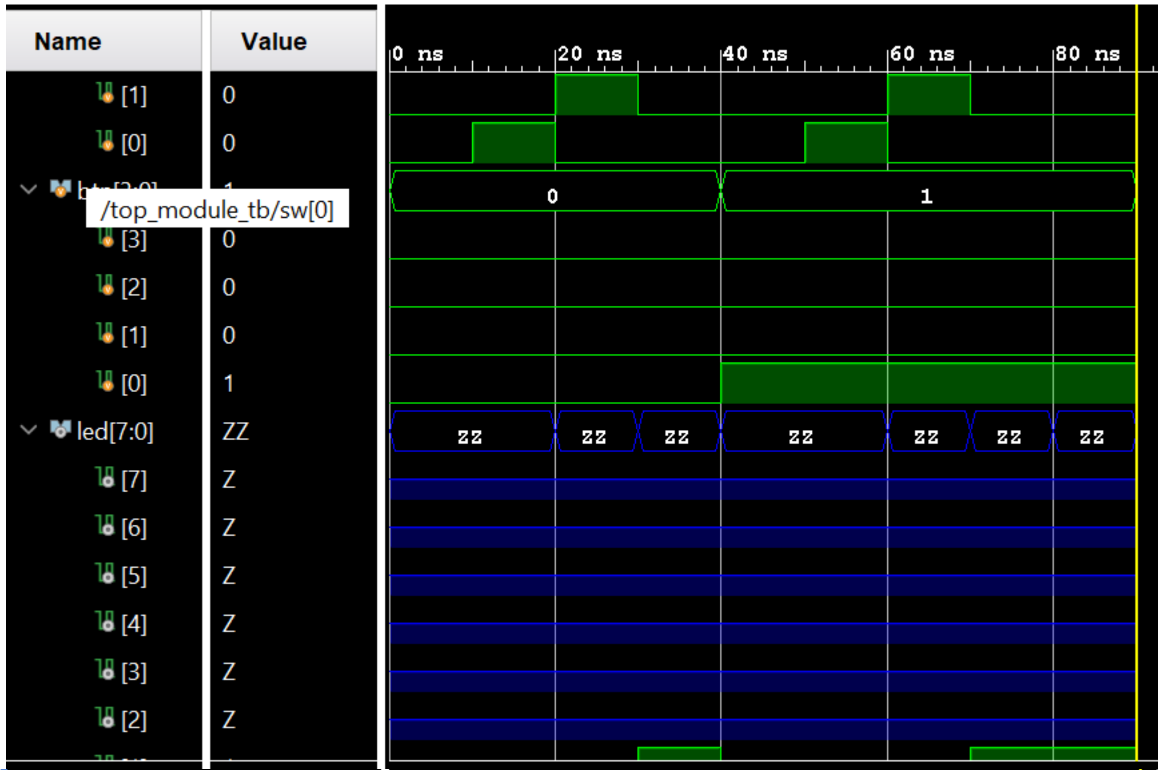
```
module top_module(
input [7:0] sw,
input [3:0] btn,
output  [7:0] led,
output  [6:0] cat,
output  [3:0] an,
output  dp
    );

    assign an =4'b1110;
    wire [15:0] new;
    assign dp = new[15];
    assign cat = new[14:8];
    assign led = new[7:0];

    // DECODER decoder1(.IN(sw[3:0]),.O(new));
```

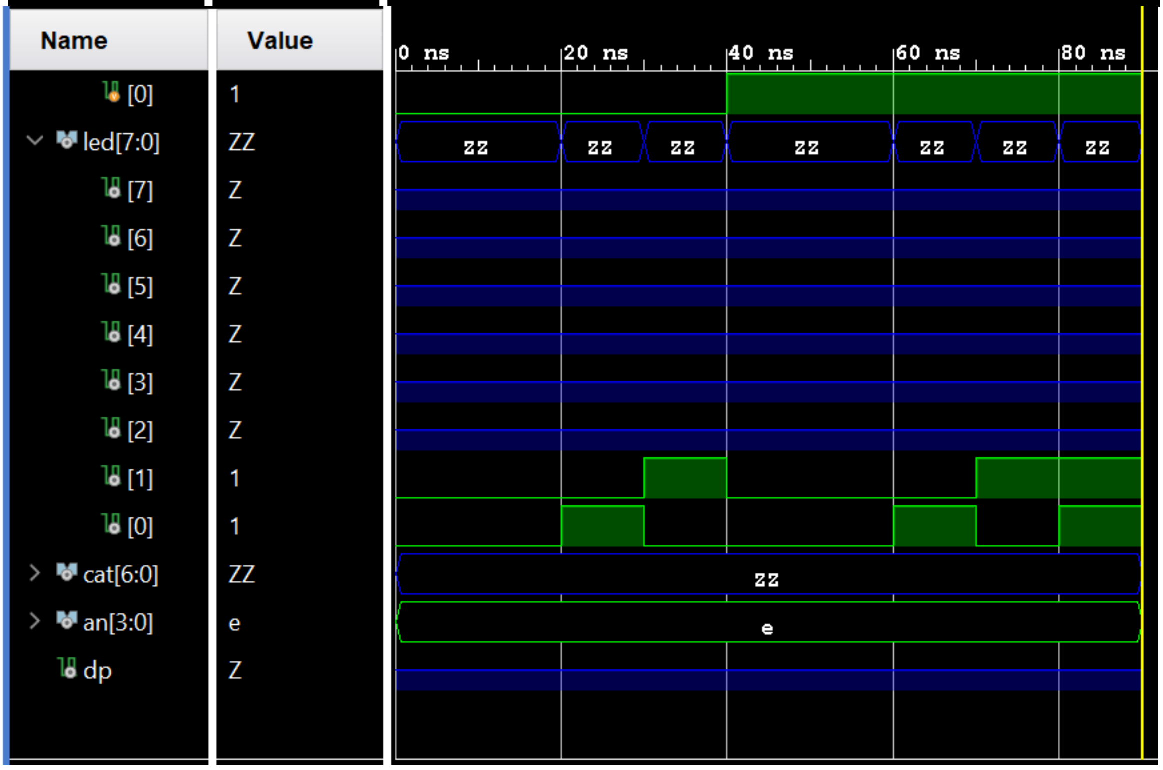
TEST BENCH



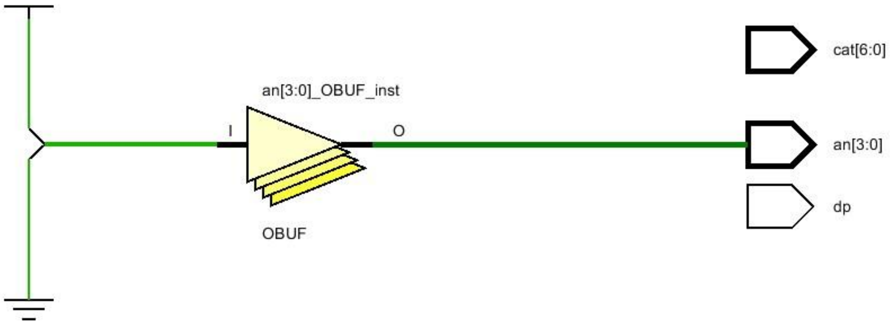
```
top_module uut(sw,btn,led,cat,an,dp);


initial
begin


   //mux
  sw=4'd0; btn=2'd0;
   #10; sw=4'b0001;
   #10; sw=4'b0010;
   #10; sw=4'b0100;
   #10; sw=4'b1000;
   ;
   sw=4'd0; btn=2'd1;
   #10; sw=4'b0001;
   #10; sw=4'b0010;
   #10; sw=4'b0100;
   #10; sw=4'b1000;
```

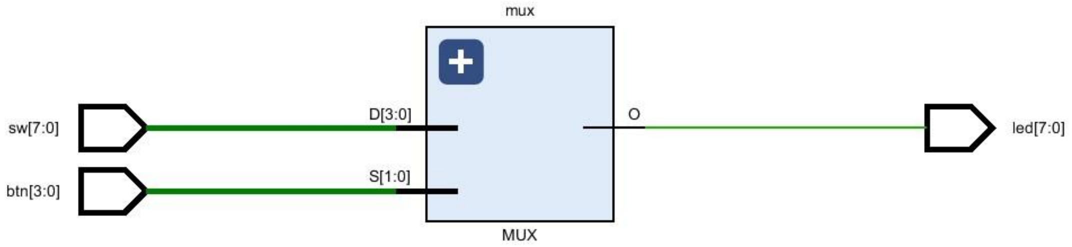
BEHAVIORAL SHEMA

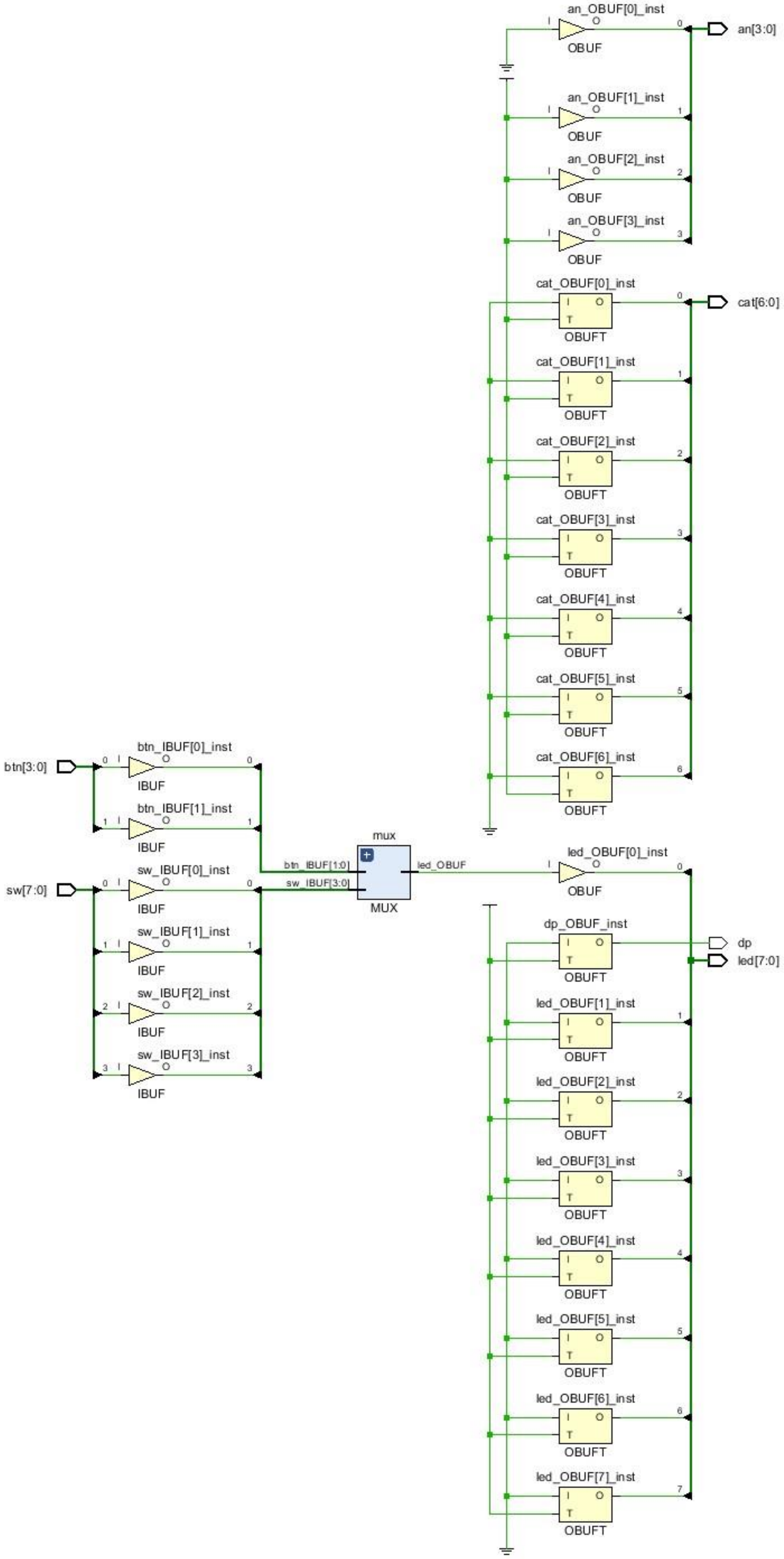




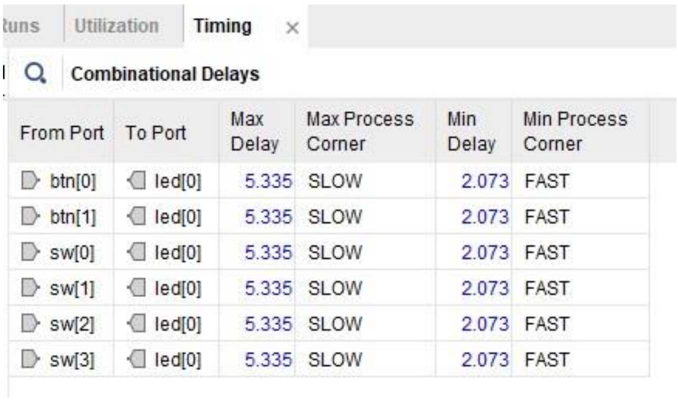
RTL SHEMA


cat[6:0]
an[3:0]_OBUF_inst
I
O
an[3:0]
dp
OBUF

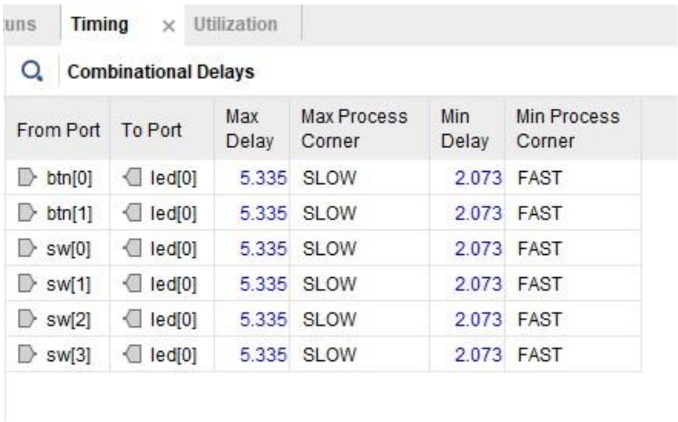

mux
sw[7:0]
D[3:0]
O
led[7:0]
btn[3:0]
S[1:0]
MUX


an_OBUF[0]_inst
OBUF
an[3:0]
an_OBUF[1]_inst
OBUF
an_OBUF[2]_inst
OBUF
an_OBUF[3]_inst
OBUF
cat_OBUF[0]_inst
OBUFT
cat[6:0]
cat_OBUF[1]_inst
OBUFT
cat_OBUF[2]_inst
OBUFT
cat_OBUF[3]_inst
OBUFT
cat_OBUF[4]_inst
OBUFT
cat_OBUF[5]_inst
OBUFT
cat_OBUF[6]_inst
OBUFT
btn[3:0]
btn_IBUF[0]_inst
IBUF
btn_IBUF[1]_inst
IBUF
sw[7:0]
sw_IBUF[0]_inst
IBUF
sw_IBUF[1]_inst
IBUF
sw_IBUF[2]_inst
IBUF
sw_IBUF[3]_inst
IBUF
mux
btn_IBUF[1:0]
sw_IBUF[3:0]
MUX
led_OBUF
led_OBUF[0]_inst
OBUF
dp_OBUF_inst
OBUFT
dp
led[7:0]
led_OBUF[1]_inst
OBUFT
led_OBUF[2]_inst
OBUFT
led_OBUF[3]_inst
OBUFT
led_OBUF[4]_inst
OBUFT
led_OBUF[5]_inst
OBUFT
led_OBUF[6]_inst
OBUFT
led_OBUF[7]_inst
OBUFT

TİMİNG COMPARE

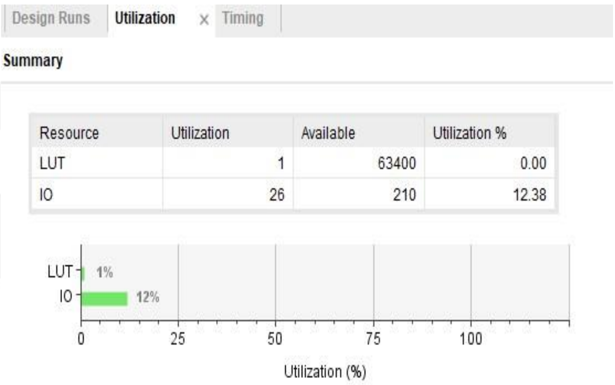


From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
btn[0]	led[0]	5.335	SLOW	2.073	FAST
btn[1]	led[0]	5.335	SLOW	2.073	FAST
sw[0]	led[0]	5.335	SLOW	2.073	FAST
sw[1]	led[0]	5.335	SLOW	2.073	FAST
sw[2]	led[0]	5.335	SLOW	2.073	FAST
sw[3]	led[0]	5.335	SLOW	2.073	FAST

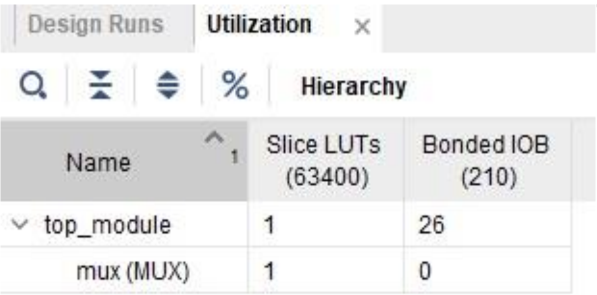


From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
btn[0]	led[0]	5.335	SLOW	2.073	FAST
btn[1]	led[0]	5.335	SLOW	2.073	FAST
sw[0]	led[0]	5.335	SLOW	2.073	FAST
sw[1]	led[0]	5.335	SLOW	2.073	FAST
sw[2]	led[0]	5.335	SLOW	2.073	FAST
sw[3]	led[0]	5.335	SLOW	2.073	FAST

UTILIZATION COMPARE



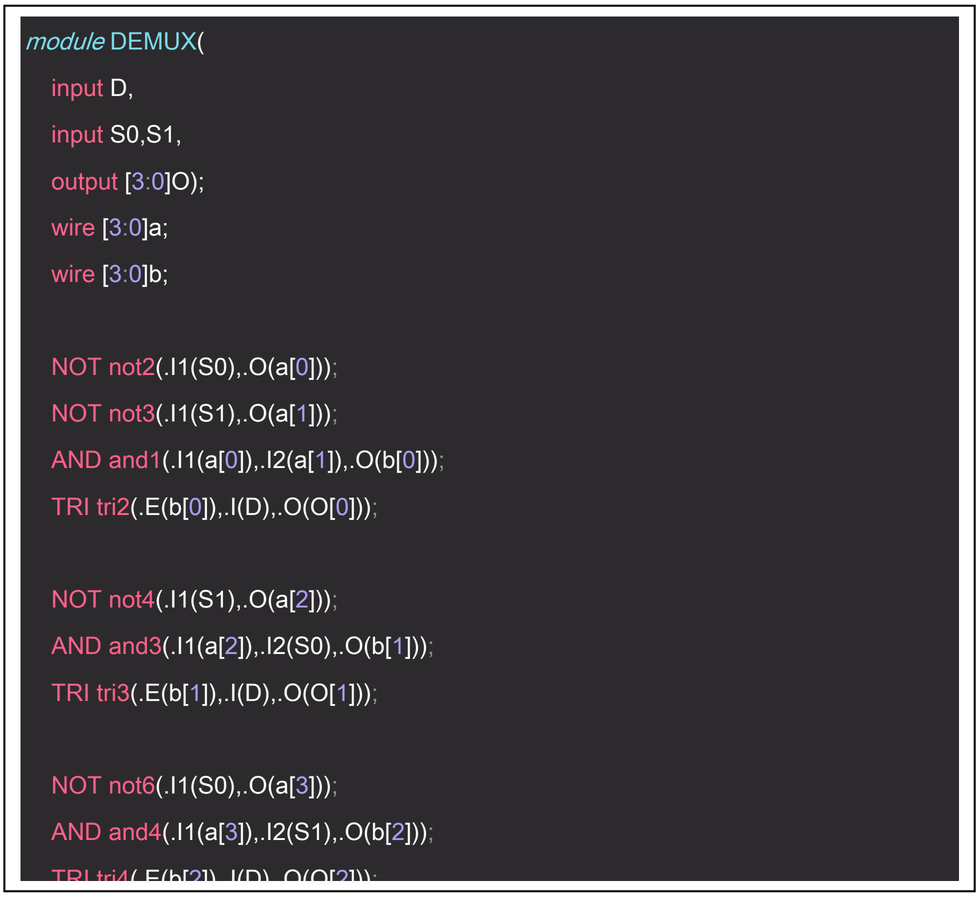
Resource	Utilization	Available	Utilization %
LUT	1	63400	0.00
IO	26	210	12.38



Name	Slice LUTs (63400)	Bonded IOB (210)
top_module	1	26
mux (MUX)	1	0

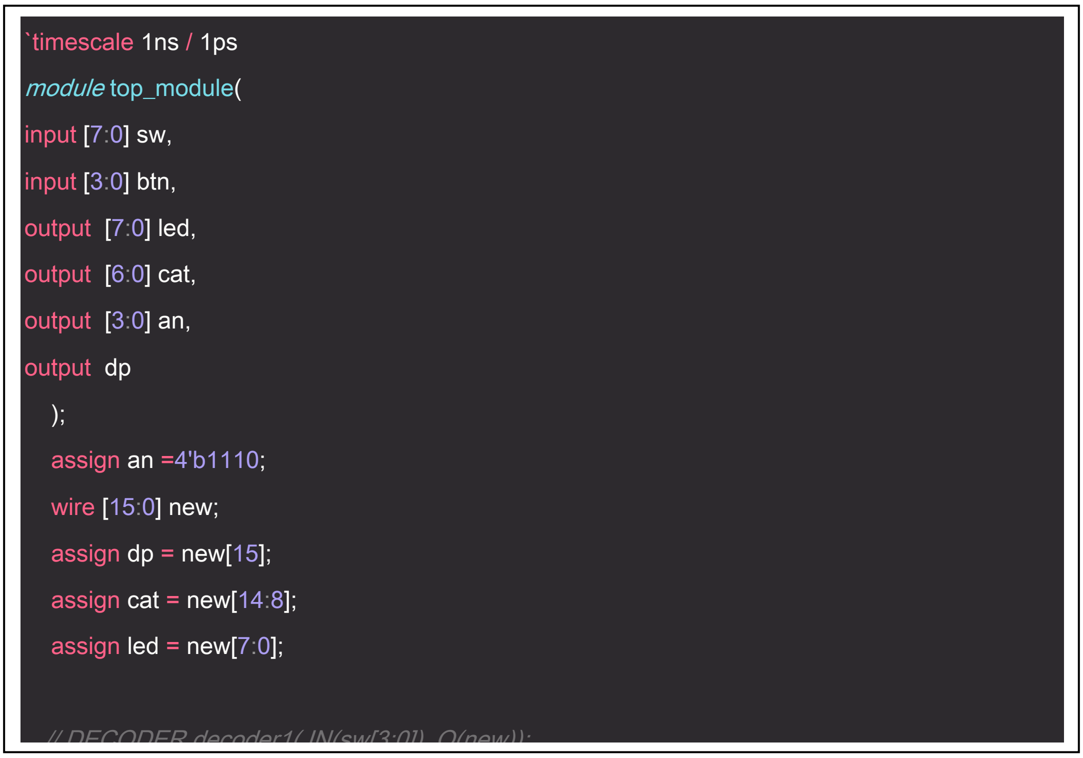
DEMULTIPLEXER

MSI LIBRARY



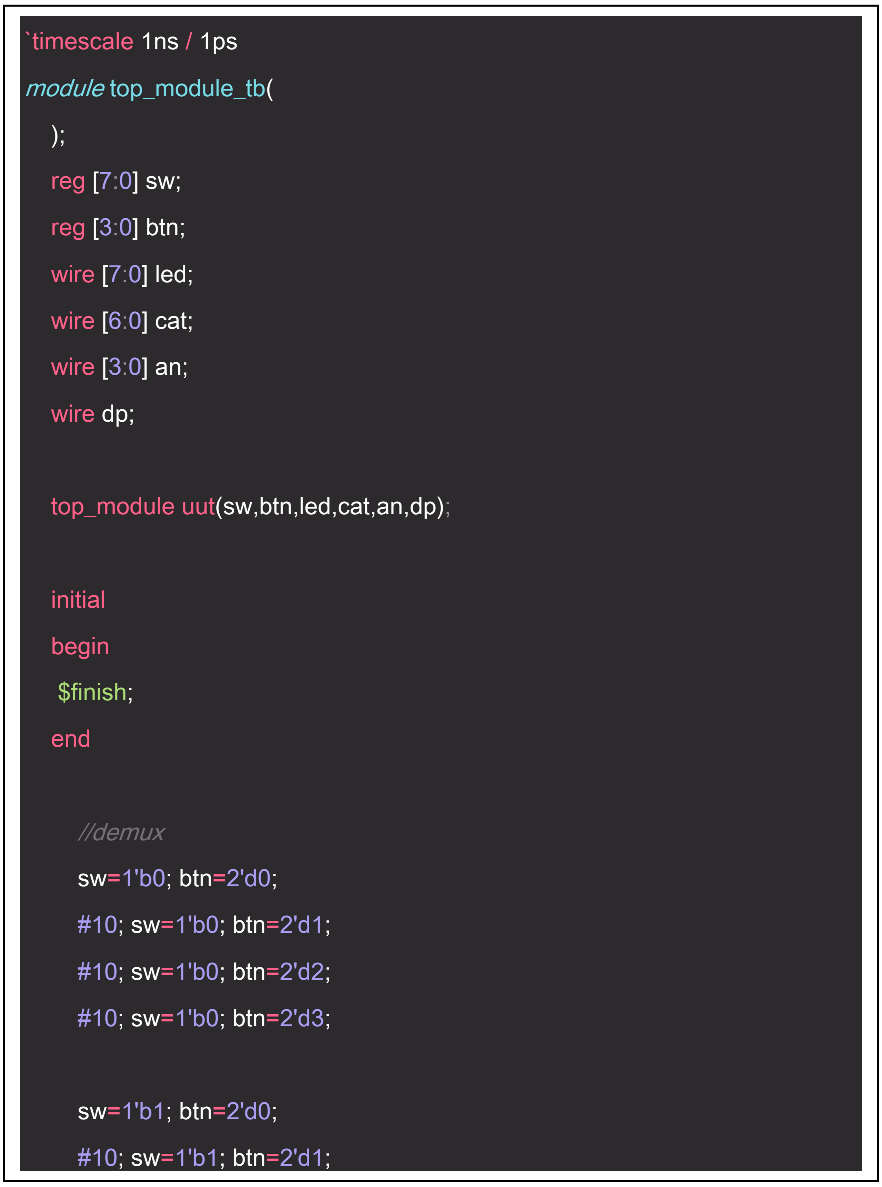
```
module DEMUX(
   input D,
   input S0,S1,
   output [3:0]O);
   wire [3:0]a;
   wire [3:0]b;

   NOT not2(.I1(S0),.O(a[0]));
   NOT not3(.I1(S1),.O(a[1]));
   AND and1(.I1(a[0]),.I2(a[1]),.O(b[0]));
   TRI tri2(.E(b[0]),.I(D),.O(O[0]));

   NOT not4(.I1(S1),.O(a[2]));
   AND and3(.I1(a[2]),.I2(S0),.O(b[1]));
   TRI tri3(.E(b[1]),.I(D),.O(O[1]));

   NOT not6(.I1(S0),.O(a[3]));
   AND and4(.I1(a[3]),.I2(S1),.O(b[2]));
   TRI tri4(.E(b[2]),.I(D),.O(O[2]));
```

TOP MODULE



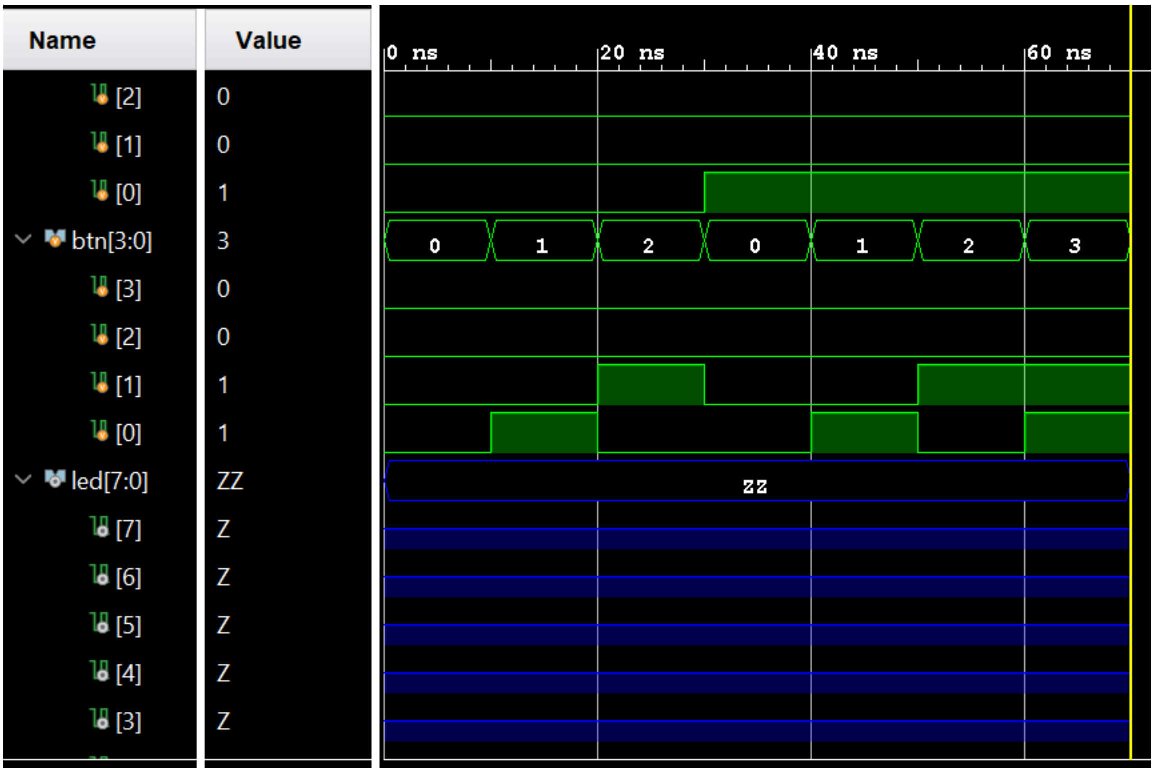
```
`timescale 1ns / 1ps
module top_module(
input [7:0] sw,
input [3:0] btn,
output  [7:0] led,
output  [6:0] cat,
output  [3:0] an,
output  dp
   );
   assign an =4'b1110;
   wire [15:0] new;
   assign dp = new[15];
   assign cat = new[14:8];
   assign led = new[7:0];


   // DECODER decoder1(.IN(sw[3:0]),.O(new));
```

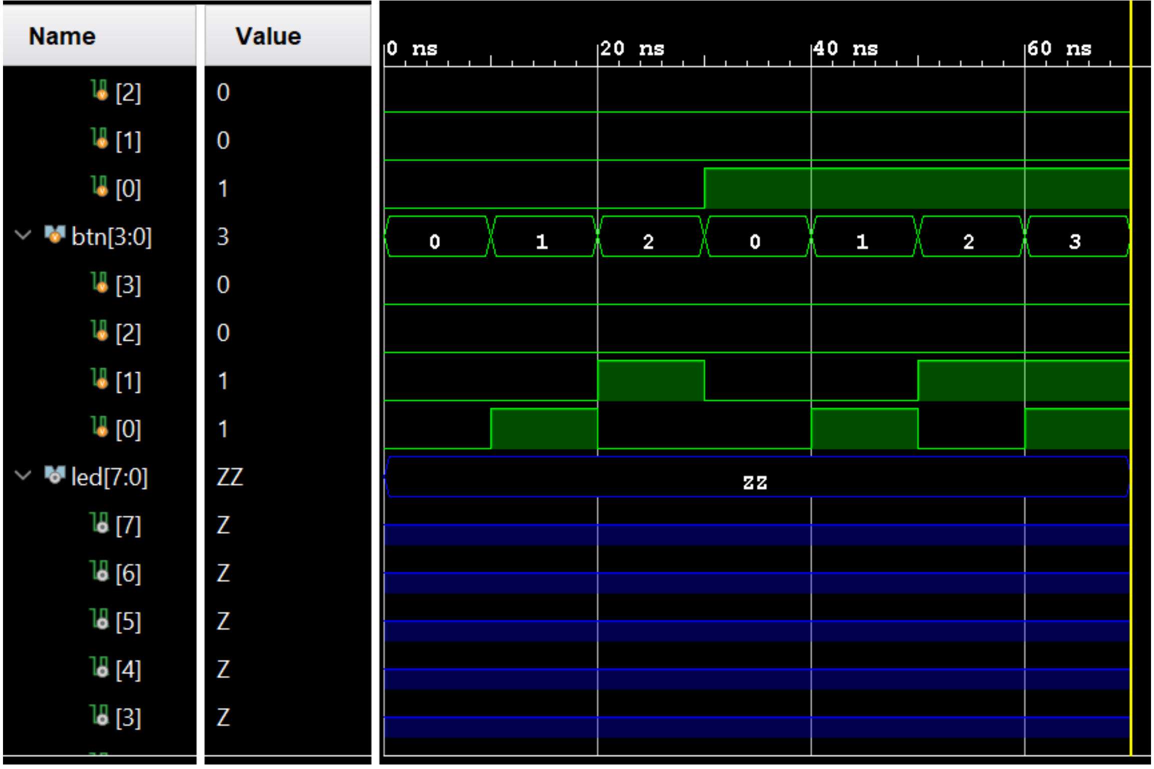
- TESTBENCH



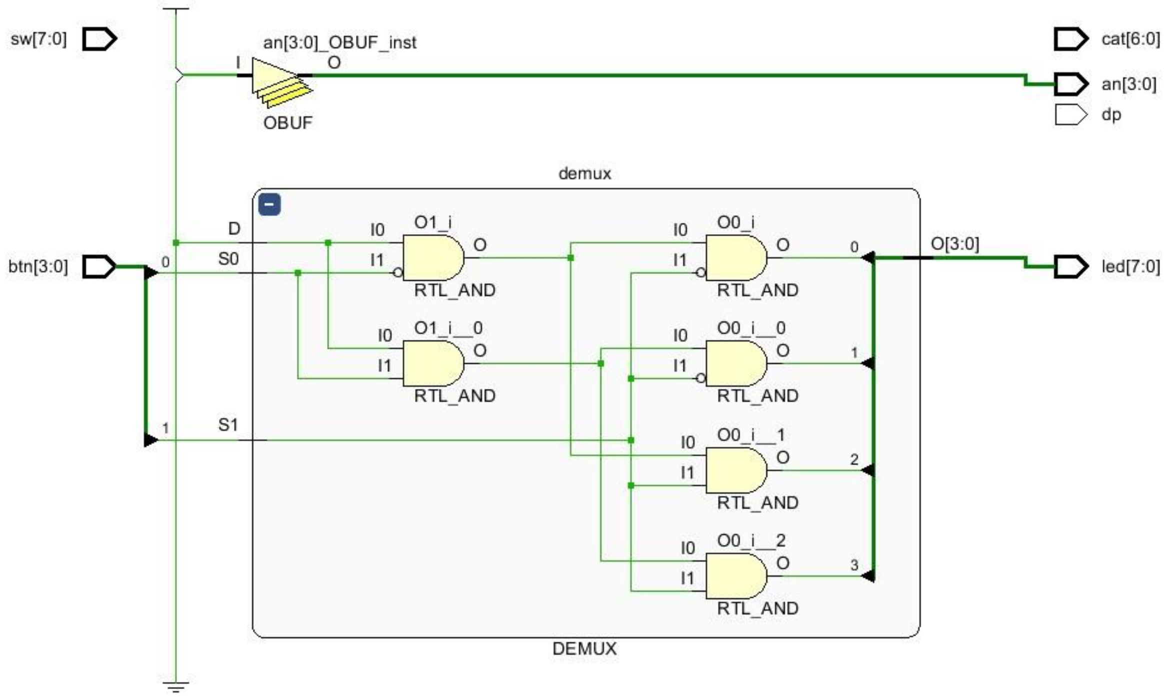
```
`timescale 1ns / 1ps
module top_module_tb(
    );
    reg [7:0] sw;
    reg [3:0] btn;
    wire [7:0] led;
    wire [6:0] cat;
    wire [3:0] an;
    wire dp;

    top_module uut(sw,btn,led,cat,an,dp);

    initial
    begin
     $finish;
    end

        //demux
        sw=1'b0; btn=2'd0;
        #10; sw=1'b0; btn=2'd1;
        #10; sw=1'b0; btn=2'd2;
        #10; sw=1'b0; btn=2'd3;

        sw=1'b1; btn=2'd0;
        #10; sw=1'b1; btn=2'd1;
```

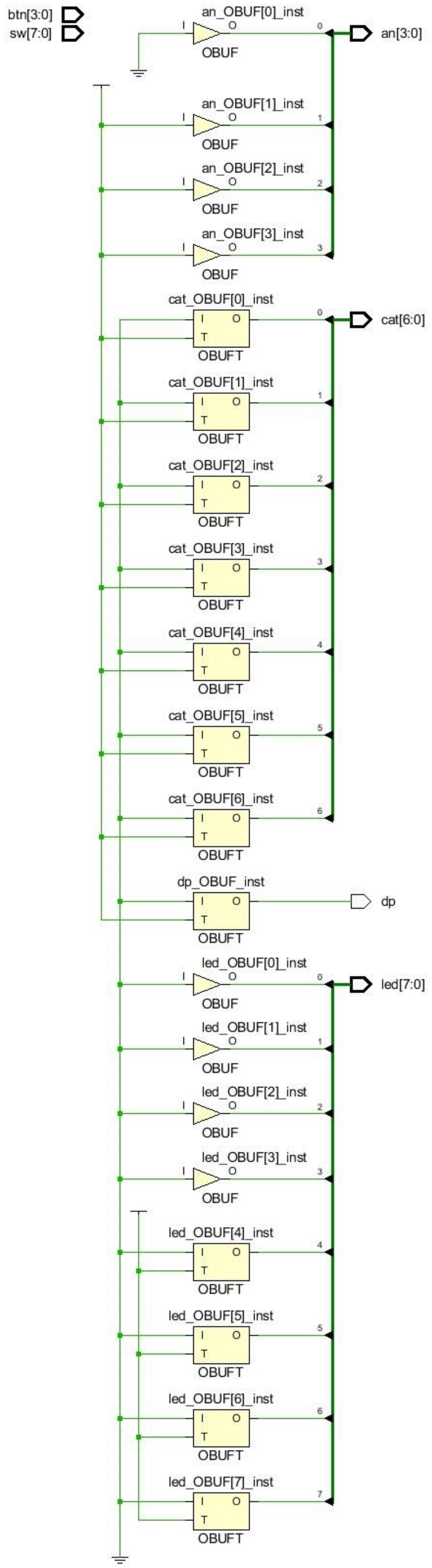
BEHAVIORAL SHEMA





RTL SHEMA


sw[7:0]
an[3:0]_OBUF_inst
I
O
OBUF
cat[6:0]
an[3:0]
dp
demux
D
S0
S1
btn[3:0]
0
1
I0
I1
O1_i
O
RTL_AND
I0
I1
O1_i__0
O
RTL_AND
I0
I1
O0_i
O
RTL_AND
I0
I1
O0_i__0
O
RTL_AND
I0
I1
O0_i__1
O
RTL_AND
I0
I1
O0_i__2
O
RTL_AND
0
1
2
3
O[3:0]
led[7:0]
DEMUX


btn[3:0]
sw[7:0]
an_OBUF[0]_inst
an_OBUF[1]_inst
an_OBUF[2]_inst
an_OBUF[3]_inst
OBUF
an[3:0]
cat_OBUF[0]_inst
cat_OBUF[1]_inst
cat_OBUF[2]_inst
cat_OBUF[3]_inst
cat_OBUF[4]_inst
cat_OBUF[5]_inst
cat_OBUF[6]_inst
OBUFT
cat[6:0]
dp_OBUF_inst
dp
led_OBUF[0]_inst
led_OBUF[1]_inst
led_OBUF[2]_inst
led_OBUF[3]_inst
led_OBUF[4]_inst
led_OBUF[5]_inst
led_OBUF[6]_inst
led_OBUF[7]_inst
led[7:0]