DIGITAL SYSTEM DESIGN APPLICATION – EHB 436E

Experiment II

Yiğit Bektaş GÜRSOY

040180063

Class Lecturer: Sıddıka Berna Örs Yalçın

Class Assistant:
Serdar Duran
Yasin Fırat Kula
Mehmet Onur Demirtürk

1. DECODER

- Truth table of a 4x16 Decoder

I_3	I_2	I_1	I_0	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	O_{12}	O_{13}	O_{14}	O_{15}
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

- DECODER Verilog code, testbench code and behavioral simulation wave screen-shots

VERILOG CODE

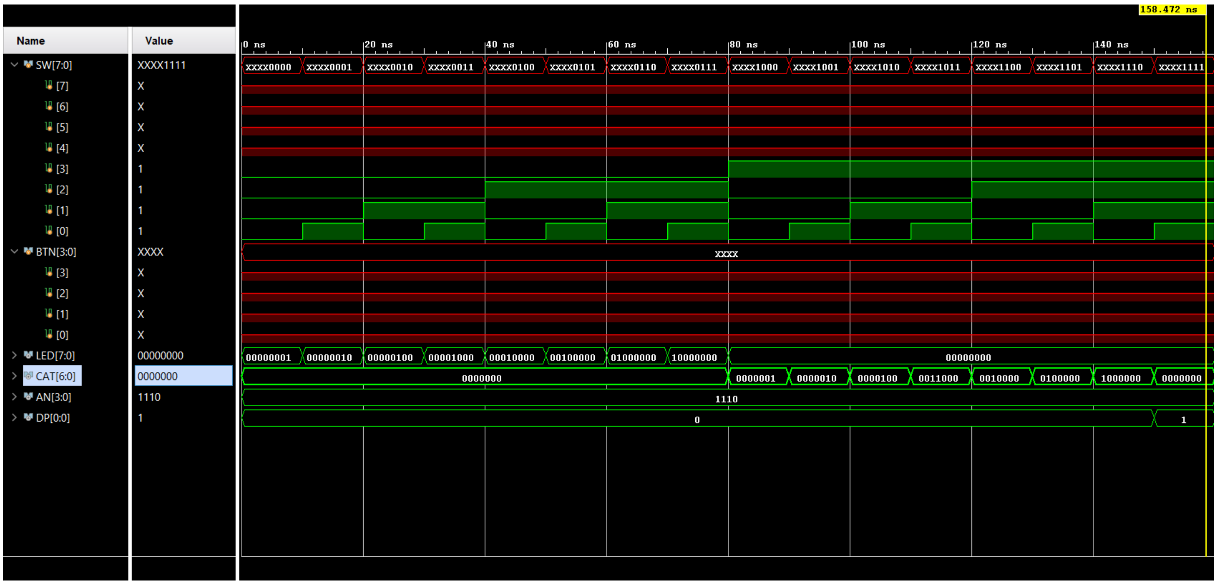
```
module DECODER (
    input [3:0]IN,
    output reg [15:0]OUT
    );
    always @(IN)
    begin
        case(IN)
            4'b0000: OUT = 16'b0000_0000_0000_0001;
            4'b0001: OUT = 16'b0000_0000_0000_0010;
            4'b0010: OUT = 16'b0000_0000_0000_0100;
            4'b0011: OUT = 16'b0000_0000_0000_1000;
            4'b0100: OUT = 16'b0000_0000_0001_0000;
            4'b0101: OUT = 16'b0000_0000_0010_0000;
            4'b0110: OUT = 16'b0000_0000_0100_0000;
            4'b0111: OUT = 16'b0000_0000_1000_0000;
            4'b1000: OUT = 16'b0000_0001_0000_0000;
            4'b1001: OUT = 16'b0000_0010_0000_0000;
            4'b1010: OUT = 16'b0000_0100_0000_0000;
            4'b1011: OUT = 16'b0000_1000_0000_0000;
            4'b1100: OUT = 16'b0001_0000_0000_0000;
            4'b1101: OUT = 16'b0010_0000_0000_0000;
            4'b1110: OUT = 16'b0100_0000_0000_0000;
            4'b1111: OUT = 16'b1000_0000_0000_0000;
        endcase
    end
endmodule
```

```
module Top_Module(
    input [7:0]SW,
    input [3:0]BTN,
    output [7:0]LED,
    output [6:0]CAT,
    output [3:0]AN,
    output [0:0]DP
    );

    DECODER DECODER1(
    .IN(SW[3:0]),
    .OUT({DP,CAT,LED})
    );
    assign AN = 4'b1110;
```

TEST BENCH CODE

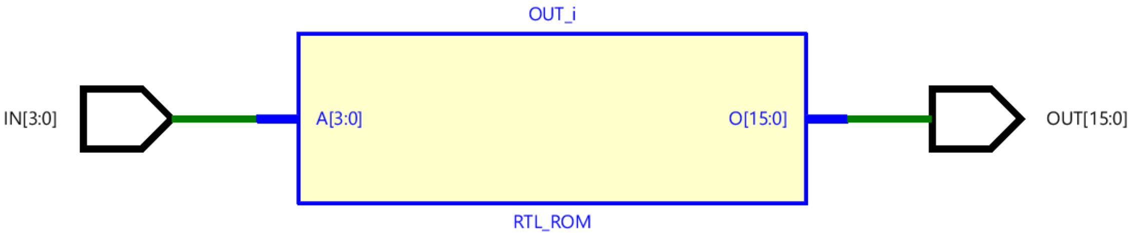
```
`timescale 1ns / 1ps

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 [0:0]DP;

    Top_Module DUT(
        .SW(SW),
        .BTN(BTN),
        .LED(LED),
        .CAT(CAT),
        .AN(AN),
        .DP(DP)
        );

        initial
        begin
            SW[3:0] = 4'h0;
        #10 SW[3:0] = 4'h1;
        #10 SW[3:0] = 4'h2;
        #10 SW[3:0] = 4'h3;
        #10 SW[3:0] = 4'h4;
        #10 SW[3:0] = 4'h5;
        #10 SW[3:0] = 4'h6;
        #10 SW[3:0] = 4'h7;
        #10 SW[3:0] = 4'h8;
        #10 SW[3:0] = 4'h9;
        #10 SW[3:0] = 4'hA;
        #10 SW[3:0] = 4'hB;
        #10 SW[3:0] = 4'hC;
        #10 SW[3:0] = 4'hD;
        #10 SW[3:0] = 4'hE;
        #10 SW[3:0] = 4'hF;
        #10 $finish;
        end
endmodule
```

BEHAVIORAL SIMULATION

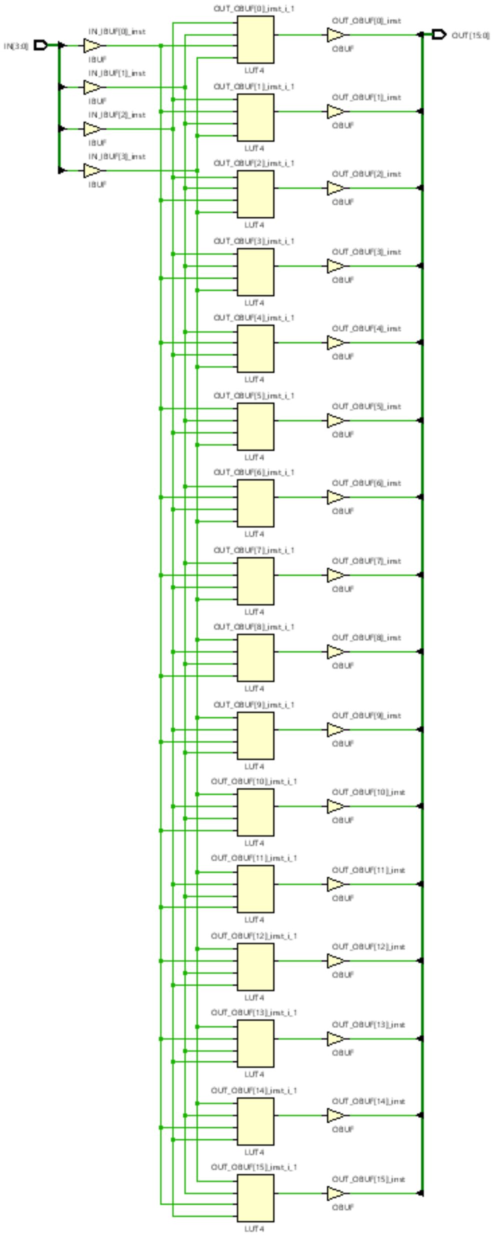


Name	Value
SW[7:0]	XXXX1111
[7]	X
[6]	X
[5]	X
[4]	X
[3]	1
[2]	1
[1]	1
[0]	1
BTN[3:0]	XXXX
[3]	X
[2]	X
[1]	X
[0]	X
LED[7:0]	00000000
CAT[6:0]	0000000
AN[3:0]	1110
DP[0:0]	1

- RTL schematic

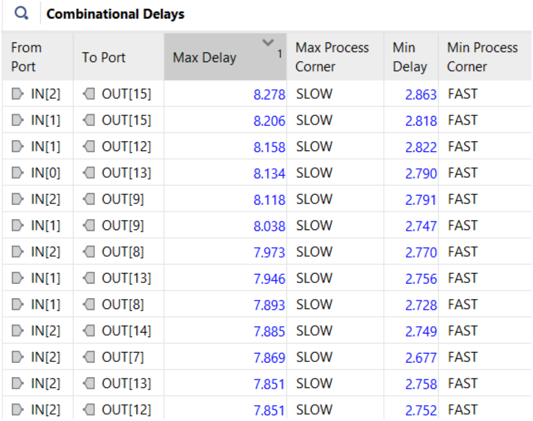


- Technology Schematic



➢ As seen in the technology schematic, we have 16 LUT4s. There are 16 different logic expressions in the diagram. Except for the bit that corresponds to the output of LUT4, the other inputs take the value 0 (corresponding to 0101 ==> 5th output). Under these conditions, LUT4 gets 1 in its logical expression. Combining LUTs this way creates the decoder.

- Greates Delay of Decoder

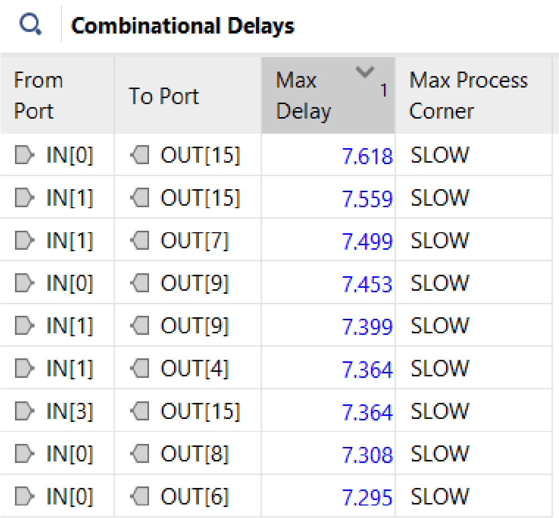


Combinational Delays

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
IN[2]	OUT[15]	8.278	SLOW	2.863	FAST
IN[1]	OUT[15]	8.206	SLOW	2.818	FAST
IN[1]	OUT[12]	8.158	SLOW	2.822	FAST
IN[0]	OUT[13]	8.134	SLOW	2.790	FAST
IN[2]	OUT[9]	8.118	SLOW	2.791	FAST
IN[1]	OUT[9]	8.038	SLOW	2.747	FAST
IN[2]	OUT[8]	7.973	SLOW	2.770	FAST
IN[1]	OUT[13]	7.946	SLOW	2.756	FAST
IN[1]	OUT[8]	7.893	SLOW	2.728	FAST
IN[2]	OUT[14]	7.885	SLOW	2.749	FAST
IN[2]	OUT[7]	7.869	SLOW	2.677	FAST
IN[2]	OUT[13]	7.851	SLOW	2.758	FAST
IN[2]	OUT[12]	7.851	SLOW	2.752	FAST

➢ Greatest delay of the implemented design is 8.278ns.

- Greatest Delay of Decoder After Timing Constrait



Combinational Delays

From Port	To Port	Max Delay	Max Process Corner
IN[0]	OUT[15]	7.618	SLOW
IN[1]	OUT[15]	7.559	SLOW
IN[1]	OUT[7]	7.499	SLOW
IN[0]	OUT[9]	7.453	SLOW
IN[1]	OUT[9]	7.399	SLOW
IN[1]	OUT[4]	7.364	SLOW
IN[3]	OUT[15]	7.364	SLOW
IN[0]	OUT[8]	7.308	SLOW
IN[0]	OUT[6]	7.295	SLOW

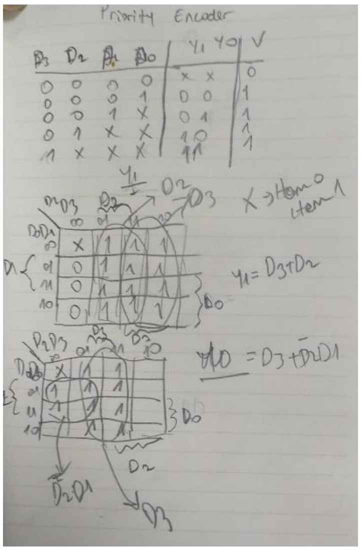
➢ Instead of the 10ns delay written on the leaflet, I observed 8.278ns delay. Then I set the maximum delay to 8ns from the edit timing constraint part. The relevant results are indicated in the photo above.

2. PRIORITY ENCODER

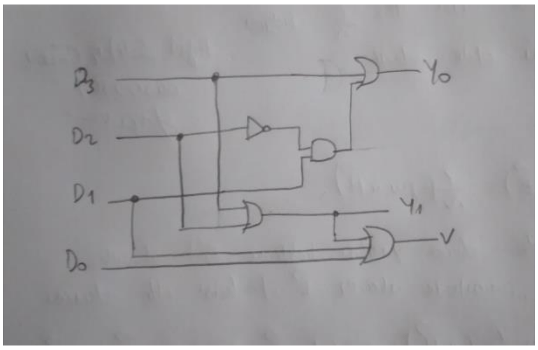
- Truth Table

D_3	D_2	D_1	D_0	OUT_0	OUT_1	V
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

- Karnaugh Map



- Hand Drawn Schematic

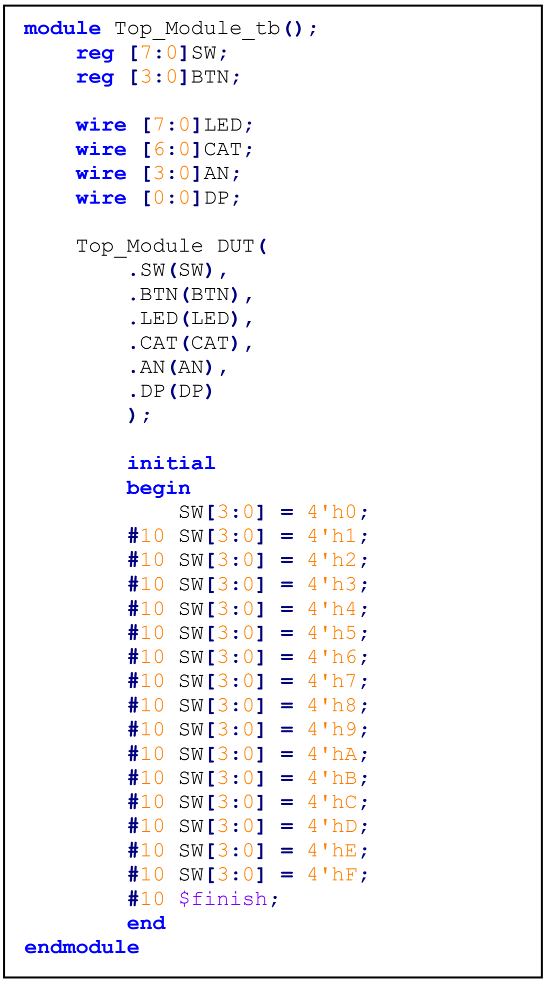


VERILOG CODE

```
module ENCODER(
    input  [3:0]IN,
    output  [1:0]OUT,
    output  [0:0]V
    );
    assign OUT[0] = (IN[3]) | (IN[1] &
~(IN[2]));
    assign OUT[1] = IN[3] | IN[2];
    assign V = (IN[0]) | (IN[1]) |
(IN[2]) | (IN[3]);

endmodule
```

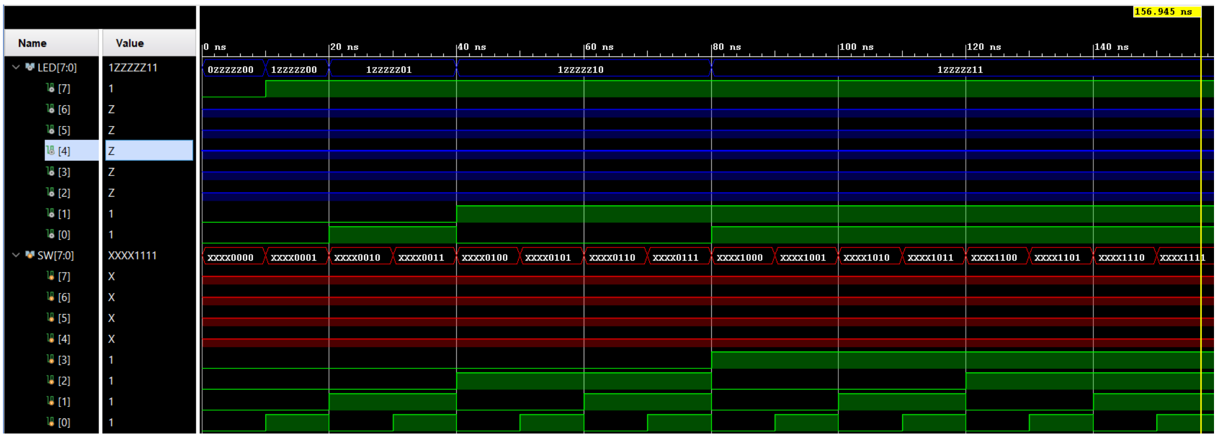
```
module Top_Module(
    input [7:0]SW,
    input [3:0]BTN,
    output [7:0]LED,
    output [6:0]CAT,
    output [3:0]AN,
    output [0:0]DP
    );
    ENCODER ENCODER1(
    .IN(SW[3:0]),
    .OUT(LED[1:0]),
    .V(LED[7])
    );
```

TESTBENCH CODE



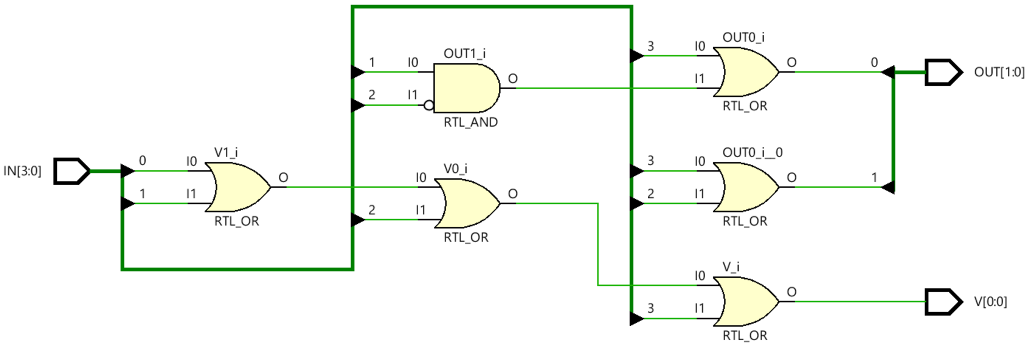
```
module Top_Module_tb();
    reg [7:0]SW;
    reg [3:0]BTN;

    wire [7:0]LED;
    wire [6:0]CAT;
    wire [3:0]AN;
    wire [0:0]DP;

    Top_Module DUT(
        .SW(SW),
        .BTN(BTN),
        .LED(LED),
        .CAT(CAT),
        .AN(AN),
        .DP(DP)
        );

        initial
        begin
            SW[3:0] = 4'h0;
        #10 SW[3:0] = 4'h1;
        #10 SW[3:0] = 4'h2;
        #10 SW[3:0] = 4'h3;
        #10 SW[3:0] = 4'h4;
        #10 SW[3:0] = 4'h5;
        #10 SW[3:0] = 4'h6;
        #10 SW[3:0] = 4'h7;
        #10 SW[3:0] = 4'h8;
        #10 SW[3:0] = 4'h9;
        #10 SW[3:0] = 4'hA;
        #10 SW[3:0] = 4'hB;
        #10 SW[3:0] = 4'hC;
        #10 SW[3:0] = 4'hD;
        #10 SW[3:0] = 4'hE;
        #10 SW[3:0] = 4'hF;
        #10 $finish;
        end
endmodule
```

BEHAVIORAL SIMULATION

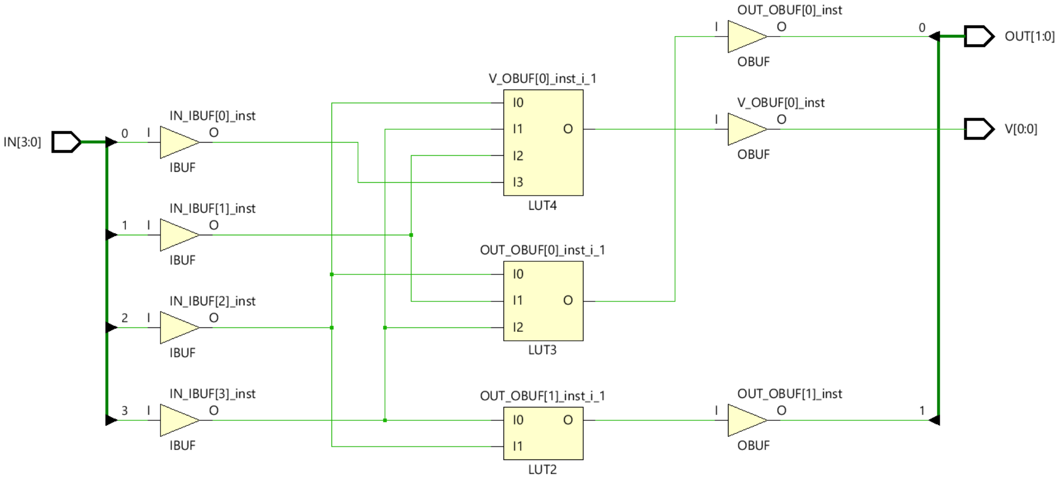


Name	Value
LED[7:0]	1ZZZZZ11
[7]	1
[6]	Z
[5]	Z
[4]	Z
[3]	Z
[2]	Z
[1]	1
[0]	1
SW[7:0]	XXXX1111
[7]	X
[6]	X
[5]	X
[4]	X
[3]	1
[2]	1
[1]	1
[0]	1

- RTL schematic



- Technology schematic



- ➢ 5 or gate 1 and gate and 1 note gate are used as seen in the RTL schematic. Total of 7 transactions were made.

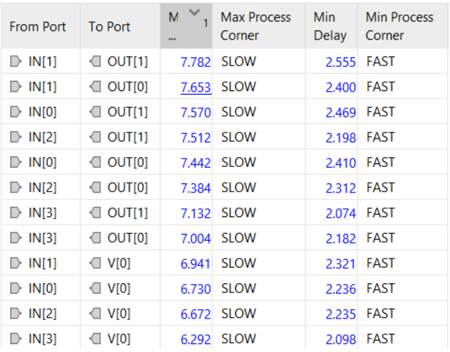
- ➢ In the technology schematic, LUT4, LUT3 and LUT2 are used. They were accompanied by 4 buffers at their entrances and 3 buffers at their exits. A total of 7 buffers were used.

VERILOG CODE

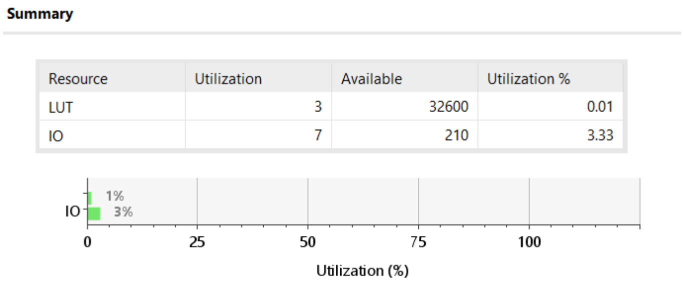
(Case Structure of Encoder)

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

        always@(IN)
    begin
        casez (IN)
            4'b0000:
            begin
                OUT = 2'b??;
                V = 1'b0;
            end

            4'b0001:
            begin
                OUT = 2'b00;
                V = 1'b1;
            end

            4'b001?:
            begin
                OUT = 2'b01;
                V = 1'b1;
            end

            4'b01??:
            begin
                OUT = 2'b10;
                V = 1'b1;
            end

            4'b1???:
            begin
                OUT = 2'b11;
                V = 1'b1;
            end

        endcase
    end
endmodule
```

- Implementation results
  - Timing and utilization report of the behavioral design.



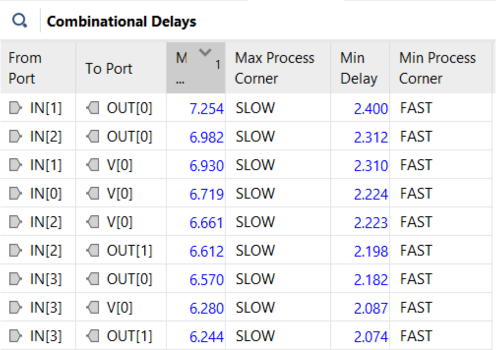
From Port	To Port	M...	Max Process Corner	Min Delay	Min Process Corner
IN[1]	OUT[1]	7.782	SLOW	2.555	FAST
IN[1]	OUT[0]	7.653	SLOW	2.400	FAST
IN[0]	OUT[1]	7.570	SLOW	2.469	FAST
IN[2]	OUT[1]	7.512	SLOW	2.198	FAST
IN[0]	OUT[0]	7.442	SLOW	2.410	FAST
IN[2]	OUT[0]	7.384	SLOW	2.312	FAST
IN[3]	OUT[1]	7.132	SLOW	2.074	FAST
IN[3]	OUT[0]	7.004	SLOW	2.182	FAST
IN[1]	V[0]	6.941	SLOW	2.321	FAST
IN[0]	V[0]	6.730	SLOW	2.236	FAST
IN[2]	V[0]	6.672	SLOW	2.235	FAST
IN[3]	V[0]	6.292	SLOW	2.098	FAST



Summary

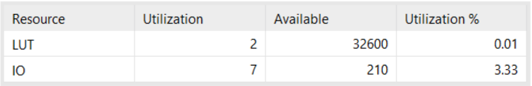
Resource	Utilization	Available	Utilization %
LUT	3	32600	0.01
IO	7	210	3.33

- Timing and utilization report of the structural design.

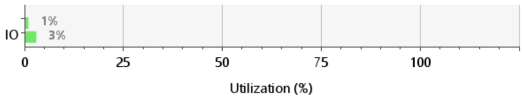


Combinational Delays

From Port	To Port	M...	Max Process Corner	Min Delay	Min Process Corner
IN[1]	OUT[0]	7.254	SLOW	2.400	FAST
IN[2]	OUT[0]	6.982	SLOW	2.312	FAST
IN[1]	V[0]	6.930	SLOW	2.310	FAST
IN[0]	V[0]	6.719	SLOW	2.224	FAST
IN[2]	V[0]	6.661	SLOW	2.223	FAST
IN[2]	OUT[1]	6.612	SLOW	2.198	FAST
IN[3]	OUT[0]	6.570	SLOW	2.182	FAST
IN[3]	V[0]	6.280	SLOW	2.087	FAST
IN[3]	OUT[1]	6.244	SLOW	2.074	FAST



Resource	Utilization	Available	Utilization %
LUT	2	32600	0.01
IO	7	210	3.33



- There is a serious difference between the field uses that are not suitable for general use. It performs better in behavioral designs than in delay performances. They have less delays than behavioral designs.

3. MULTIPLEXER

VERILOG CODE

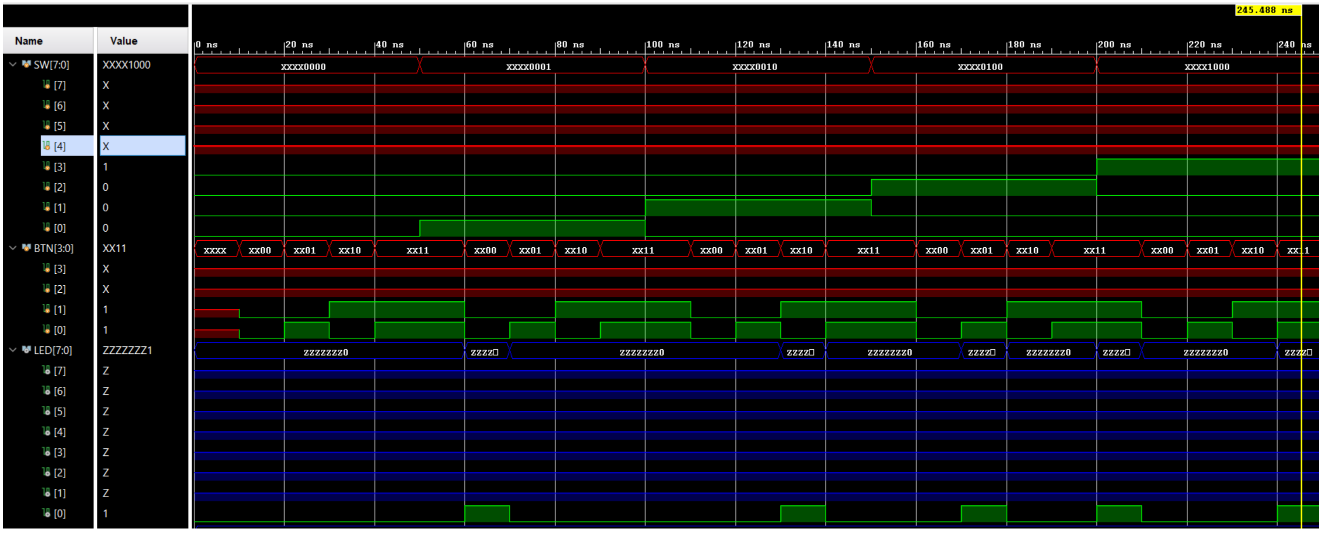
```
module MUX(
    input [3:0]D,
    input [1:0]S,
    output reg [0:0]O);

    wire y0,y1,y2,y3;

    assign y0 = D[0] && ~S[0] &&
~S[1];
    assign y1 = D[1] && ~S[0] &&
S[1];
    assign y2 = D[2] && S[0] &&
~S[1];
    assign y3 = D[3] && S[0] &&
S[1];

    assign O = y0 || y1 || y2 ||
y3;

endmodule
```

```
module Top_Module(
    input [7:0]SW,
    input [3:0]BTN,
    output [7:0]LED,
    output [6:0]CAT,
    output [3:0]AN,
    output [0:0]DP
    );

 MUX MUX1(
    .D(SW[3:0]),
    .S(BTN[1:0]),
    .O(LED[0]));
```

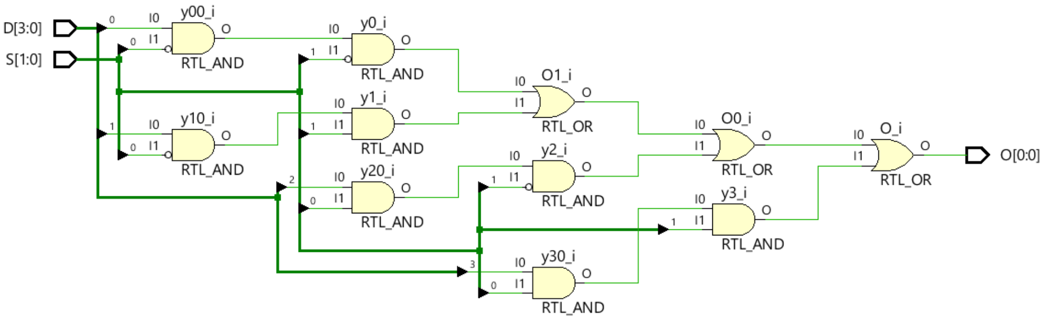
TESTBENCH CODE

```
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 [0:0]DP;

    Top_Module DUT(
        .SW(SW),
        .BTN(BTN),
        .LED(LED),
        .CAT(CAT),
        .AN(AN),
        .DP(DP)
        );

        initial
        begin
            SW[3:0] = 4'b0000; #10 BTN[2:0] = 2'h0 ;#10  BTN[2:0] = 2'h1 ;#10  BTN[2:0] = 2'h2 ;#10  BTN[2:0] = 2'h3 ;
        #10 SW[3:0] = 4'b0001; #10 BTN[2:0] = 2'h0 ;#10  BTN[2:0] = 2'h1 ;#10  BTN[2:0] = 2'h2 ;#10  BTN[2:0] = 2'h3 ;
        #10 SW[3:0] = 4'b0010; #10 BTN[2:0] = 2'h0 ;#10  BTN[2:0] = 2'h1 ;#10  BTN[2:0] = 2'h2 ;#10  BTN[2:0] = 2'h3 ;
        #10 SW[3:0] = 4'b0100; #10 BTN[2:0] = 2'h0 ;#10  BTN[2:0] = 2'h1 ;#10  BTN[2:0] = 2'h2 ;#10  BTN[2:0] = 2'h3 ;
        #10 SW[3:0] = 4'b1000; #10 BTN[2:0] = 2'h0 ;#10  BTN[2:0] = 2'h1 ;#10  BTN[2:0] = 2'h2 ;#10  BTN[2:0] = 2'h3 ;
        #10 $finish;
        end
endmodule
```

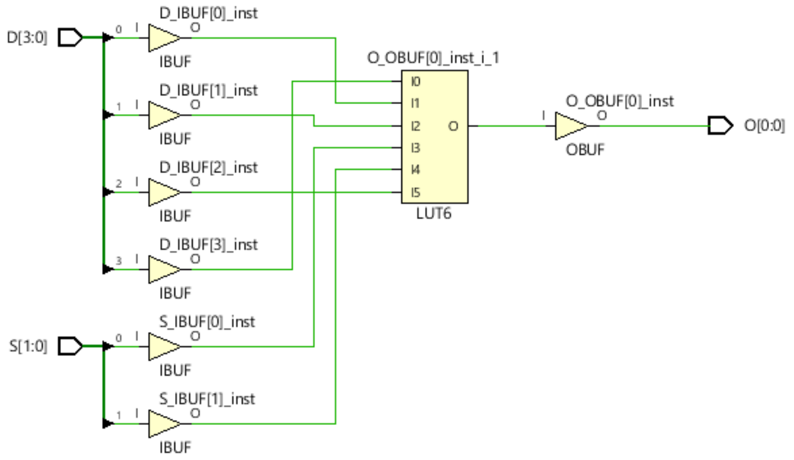
BEHAVIORAL SIMULATION



- RTL schematic



- Technology schematic

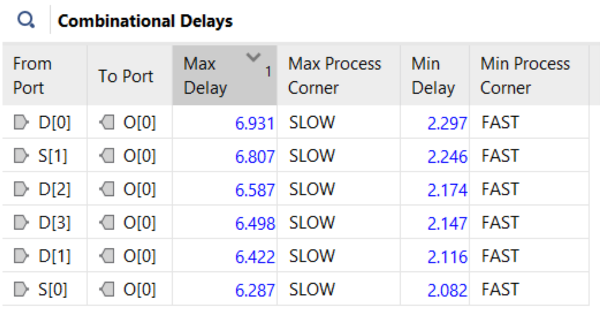


VERILOG CODE

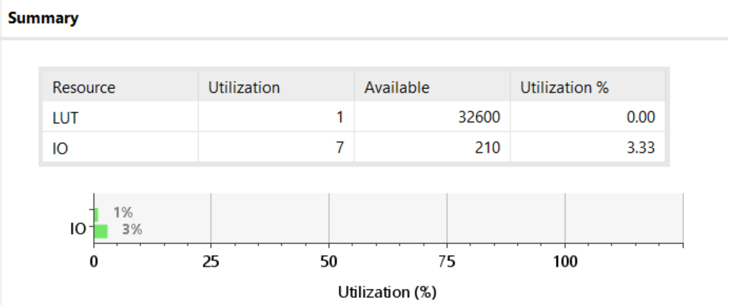
(Case Structure of MUX)

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

    always @(D,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
endmodule
```

- Implementation results
  - Timing and utilization report of the behavioral design.

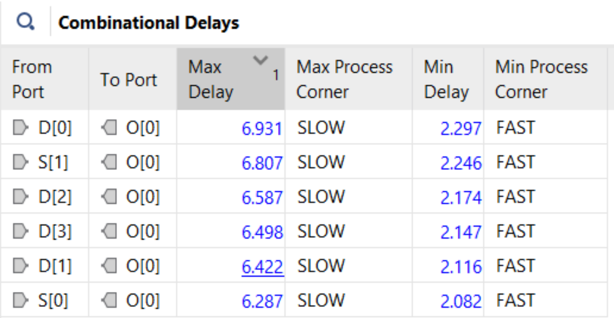

Combinational Delays

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
D[0]	O[0]	6.931	SLOW	2.297	FAST
S[1]	O[0]	6.807	SLOW	2.246	FAST
D[2]	O[0]	6.587	SLOW	2.174	FAST
D[3]	O[0]	6.498	SLOW	2.147	FAST
D[1]	O[0]	6.422	SLOW	2.116	FAST
S[0]	O[0]	6.287	SLOW	2.082	FAST


Summary

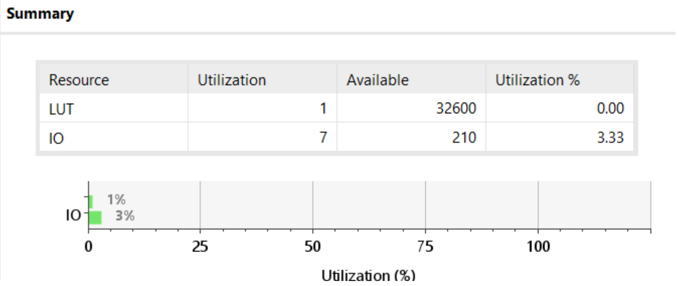
Resource	Utilization	Available	Utilization %
LUT	1	32600	0.00
IO	7	210	3.33

- Timing and utilization report of the structural design.



Combinational Delays

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
D[0]	O[0]	6.931	SLOW	2.297	FAST
S[1]	O[0]	6.807	SLOW	2.246	FAST
D[2]	O[0]	6.587	SLOW	2.174	FAST
D[3]	O[0]	6.498	SLOW	2.147	FAST
D[1]	O[0]	6.422	SLOW	2.116	FAST
S[0]	O[0]	6.287	SLOW	2.082	FAST



Summary

Resource	Utilization	Available	Utilization %
LUT	1	32600	0.00
IO	7	210	3.33

- Delay times and area are very similar in between structural and behavioral design.

4. DEMULTIPLEXER

VERILOG CODE

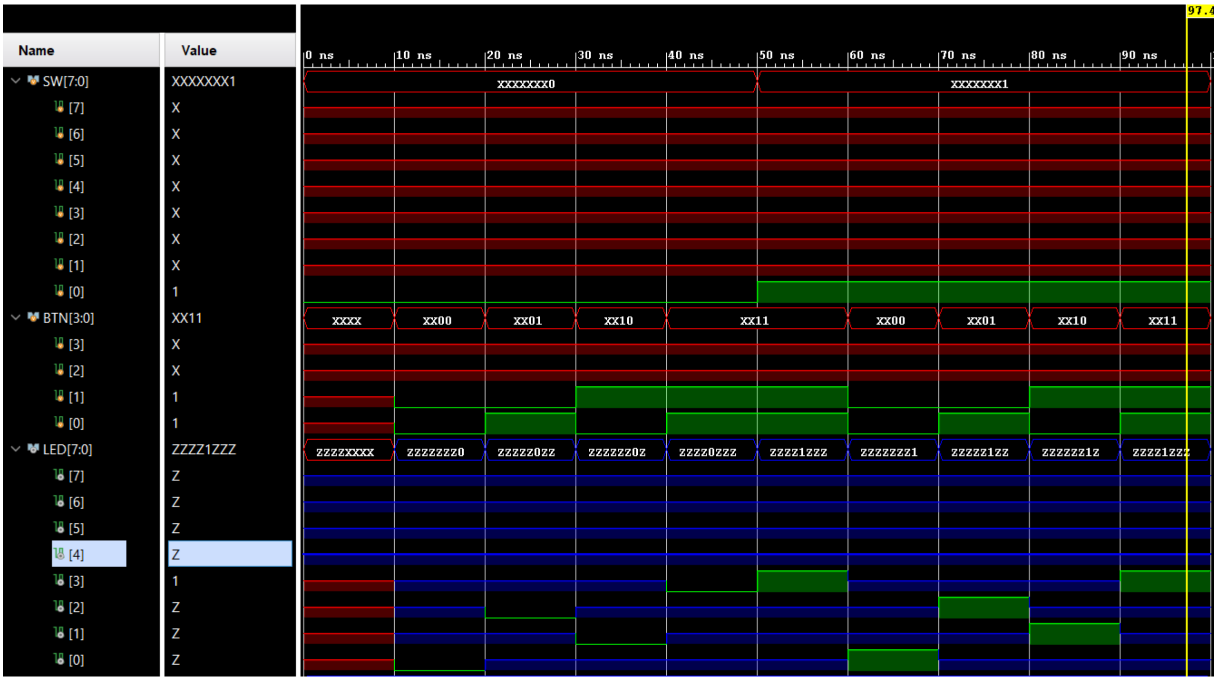
```verilog
module DEMUX(
    input [0:0]D,
    input [1:0]S,
    output [3:0]O
    );

    wire s0_not, s1_not;

    NOT_gate NOT1(.I1(S[0]), .O(s0_not));
    NOT_gate NOT2(.I1(S[1]), .O(s1_not));

    wire y0,y1,y2,y3;

    AND_gate AND1( .I1(s0_not),
.I2(s1_not), .O(y0));
    AND_gate AND2( .I1(s0_not), .I2(S[1]),
.O(y1));
    AND_gate AND3( .I1(S[0]), .I2(s1_not),
.O(y2));
    AND_gate AND4( .I1(S[0]), .I2(S[1]),
.O(y3));

    TRI TRI1( .I(D[0]), .E(y0), .O(O[0]));
    TRI TRI2( .I(D[0]), .E(y1), .O(O[1]));
    TRI TRI3( .I(D[0]), .E(y2), .O(O[2]));
    TRI TRI4( .I(D[0]), .E(y3), .O(O[3]));

endmodule
```

```verilog
module Top_Module(
    input [7:0]SW,
    input [3:0]BTN,
    output [7:0]LED,
    output [6:0]CAT,
    output [3:0]AN,
    output [0:0]DP
    );

    DEMUX DEMUX1(
    .D(SW[0]),
    .S(BTN[1:0]),
    .O(LED[3:0]));

endmodule
```

TESTBENCH CODE

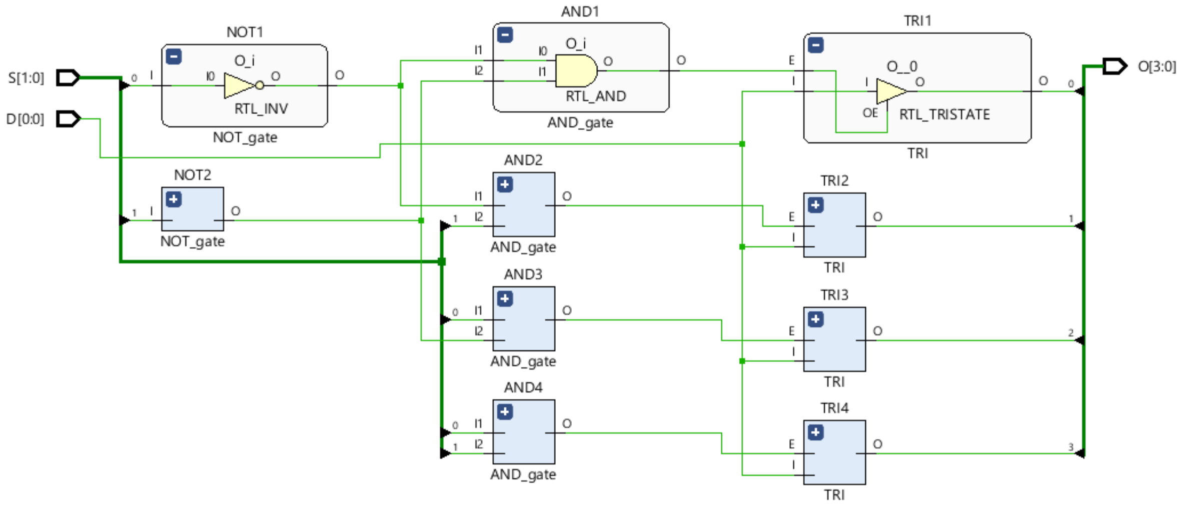
```
module Top_Module_tb();
    reg [7:0]SW;
    reg [3:0]BTN;

    wire [7:0]LED;
    wire [6:0]CAT;
    wire [3:0]AN;
    wire [0:0]DP;

    Top_Module DUT(
        .SW(SW),
        .BTN(BTN),
        .LED(LED),
        .CAT(CAT),
        .AN(AN),
        .DP(DP)
        );

        initial
        begin
            SW[0]= 1'b0; #10 BTN[1:0] = 2'h0 ;#10  BTN[1:0] = 2'h1 ;#10  BTN[1:0] = 2'h2 ;#10  BTN[1:0] = 2'h3;
        #10 SW[0]= 1'b1; #10 BTN[1:0] = 2'h0 ;#10  BTN[1:0] = 2'h1 ;#10  BTN[1:0] = 2'h2 ;#10  BTN[1:0] = 2'h3;
        #10 $finish;
        end
endmodule
```

BEHAVIORAL SIMULATION

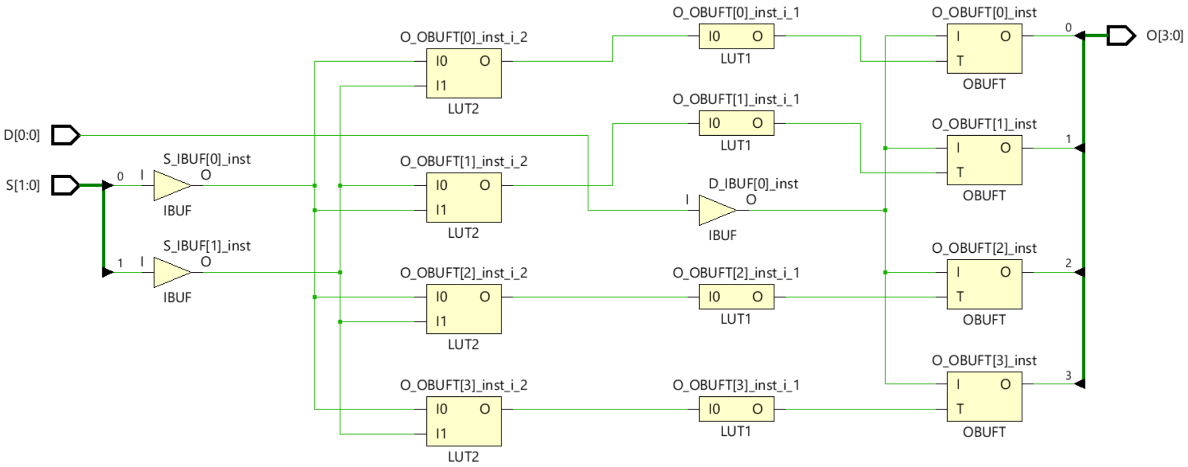


Name	Value
SW[7:0]	XXXXXXX1
[7]	X
[6]	X
[5]	X
[4]	X
[3]	X
[2]	X
[1]	X
[0]	1
BTN[3:0]	XX11
[3]	X
[2]	X
[1]	1
[0]	1
LED[7:0]	ZZZZ1ZZZ
[7]	Z
[6]	Z
[5]	Z
[4]	Z
[3]	1
[2]	Z
[1]	Z
[0]	Z

- RTL schematic



- Technology schematic


D[0:0]
S[1:0]
S_IBUF[0]_inst
IBUF
S_IBUF[1]_inst
IBUF
O_OBUFT[0]_inst_i_2
LUT2
O_OBUFT[1]_inst_i_2
LUT2
O_OBUFT[2]_inst_i_2
LUT2
O_OBUFT[3]_inst_i_2
LUT2
O_OBUFT[0]_inst_i_1
LUT1
O_OBUFT[1]_inst_i_1
LUT1
D_IBUF[0]_inst
IBUF
O_OBUFT[2]_inst_i_1
LUT1
O_OBUFT[3]_inst_i_1
LUT1
O_OBUFT[0]_inst
OBUFT
O_OBUFT[1]_inst
OBUFT
O_OBUFT[2]_inst
OBUFT
O_OBUFT[3]_inst
OBUFT
O[3:0]