Ömer Faruk SERT

040170244

SSI_library.v

```
`timescale 1ns / 1ps

module AND( output O,
            input I1,I2);

            assign O = I1&I2;
            endmodule

module OR (output O,
           input I1,I2);

           assign O = I1|I2;
           endmodule

module NOT( output O,
            input I);

            assign O = ~I;
            endmodule

module NAND(output reg O,
            input I1,I2);

            always @(I1,I2) begin
            O = ~(I1&I2);
            end
            endmodule

module NOR(output reg O,
            input I1,I2);

            always @(I1,I2) begin
                O = ~(I1|I2);
            end
            endmodule

module EXOR(output O,
            input I1,I2);
            LUT2 #(
                .INIT ( 4'b0110 )
            ) EXOR
            (
```

```
    .I0( I1 ),
    .I1( I2 ),
    .O ( O )
 );
 endmodule

module EXNOR(output O,
            input I1,I2);
            LUT2 #(
                .INIT ( 4'b1001 )
            ) EXNOR
            (
    .I0( I1 ),
    .I1( I2 ),
    .O ( O )
 );
 endmodule

 module TRI(input I,E,
            output O);
            assign O = (E==1) ? (I) : (1'bz);
endmodule
```

Top_module.v

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 10/13/2022 01:10:23 PM
// Design Name:
// Module Name: Top_module
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module Top_module(O,IN

    );
    output [7:0]O;
    input [15:0]IN;

    AND GATE1(.O(O[0]),.I1(IN[0]),.I2(IN[1]));
    OR GATE2(.O(O[1]),.I1(IN[2]),.I2(IN[3]));

    NAND GATE4(.O(O[2]),.I1(IN[4]),.I2(IN[5]));
    NOR GATE5(.O(O[3]),.I1(IN[6]),.I2(IN[7]));
    EXOR GATE6(.O(O[4]),.I1(IN[8]),.I2(IN[9]));
    EXNOR GATE7(.O(O[5]),.I1(IN[10]),.I2(IN[11]));
    TRI GATE8(.O(O[6]),.I(IN[12]),.E(IN[13]));

    NOT GATE3(.O(O[7]),.I(IN[14]));

endmodule
```

Top_module_tb.v

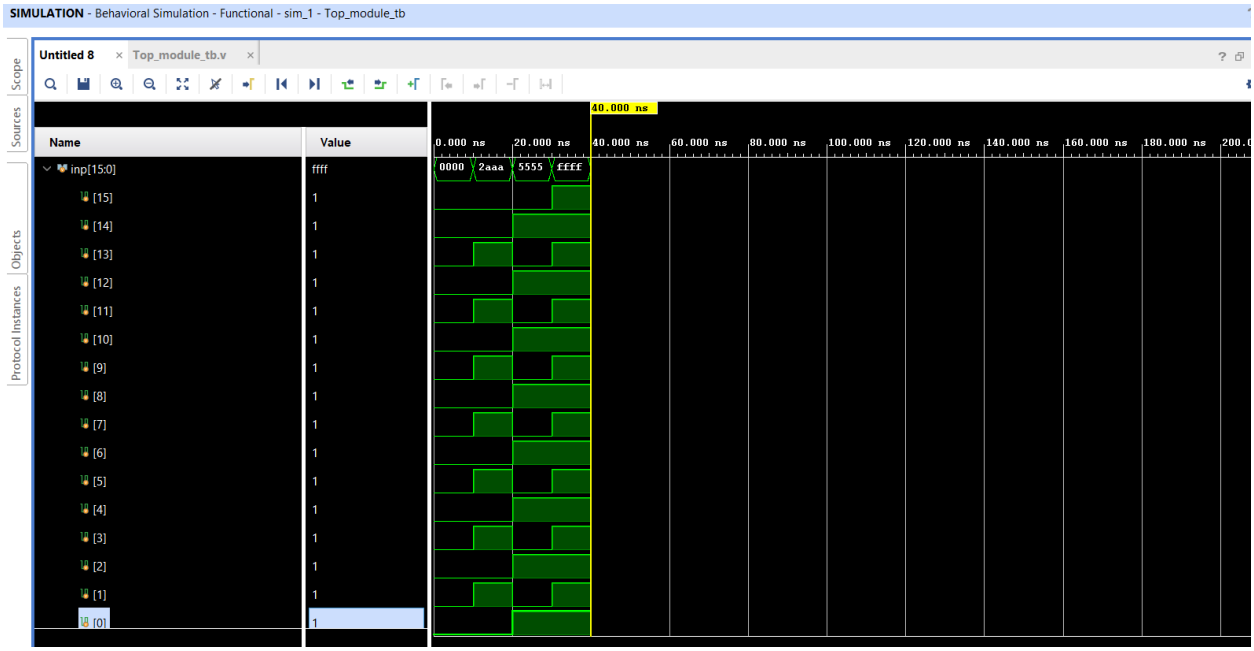
```
`timescale 1ns / 1ps


module Top_module_tb;
    reg [15:0] inp;
    wire [7:0] out;
    integer i;

    Top_module uut(.O(out), .IN(inp));

    initial
    begin
        // inp1 =0 inp2=0
    inp=16'h00;
    #10
    // inp1=0 inp2=1
    for (i=0 ; i<14 ; i=i+2 ) begin
        inp[i]=0;inp[i+1]=1;
    end
    #10
    // inp1=1; inp2=0; also changing not input from 0 to 1
    for (i=0 ; i<14 ; i=i+2 ) begin
        inp[i]=1;inp[i+1]=0;
    end
    inp[14]=1;
    #10
    inp=16'b1111111111111111;
    #10
    $finish;

    end
endmodule
```



Input 0,1 and

Input 2,3 or

Input 4,5 nand

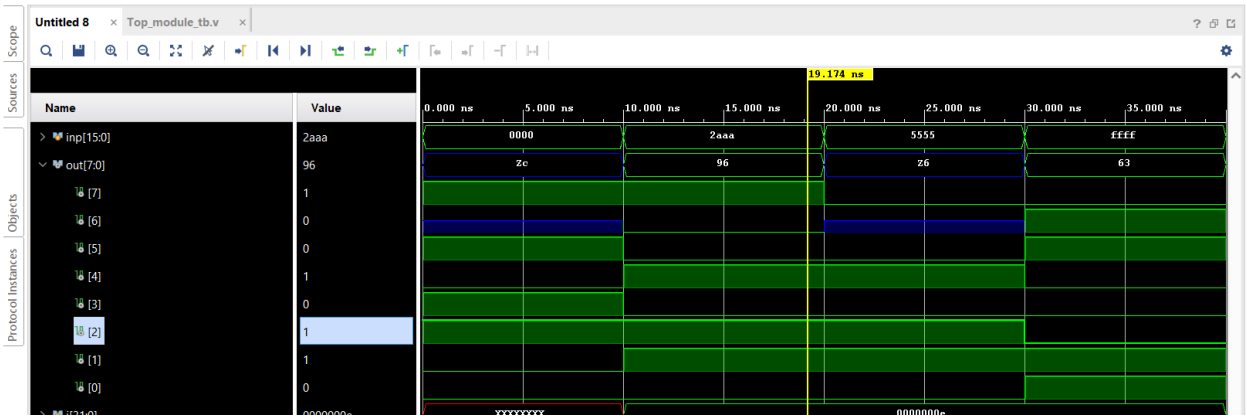
Input 6,7 nor

Input 8,9 exor

Input 10,11 exnor

Input 12,13 tri 13 is E port

Input 14 not



Output 0 and

Output 1 or

Output 2 nand

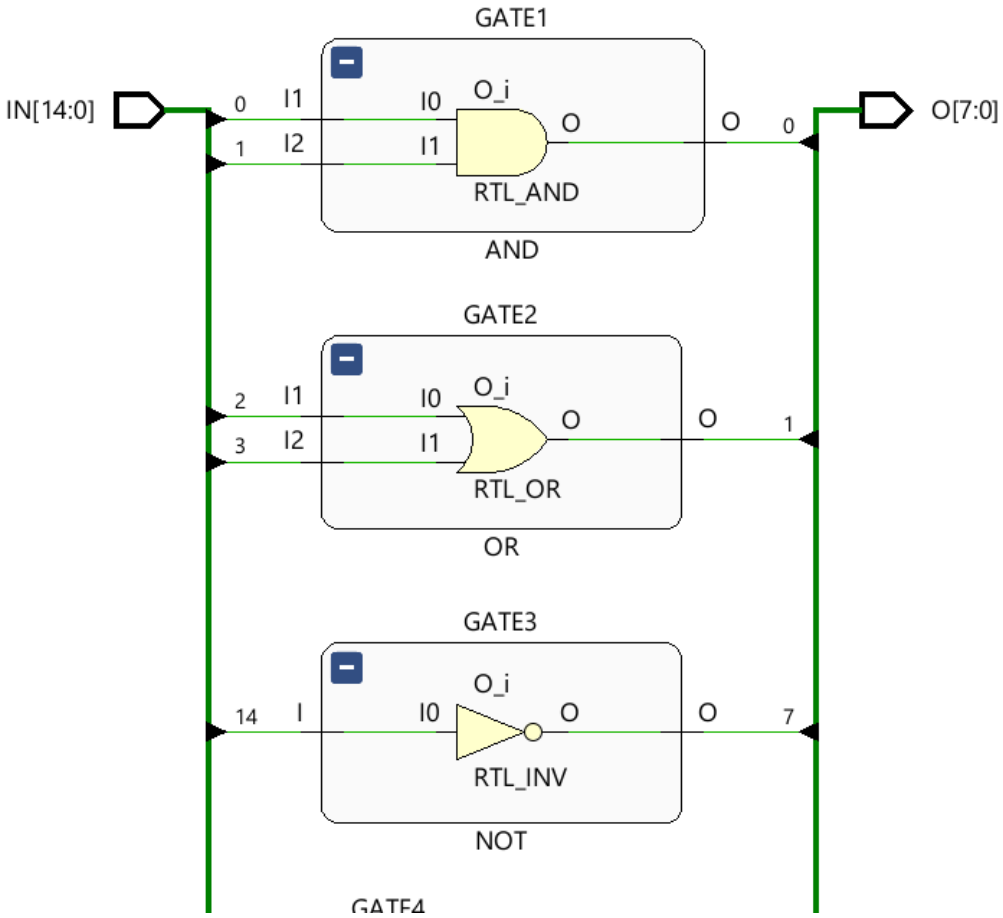
Output 3 nor

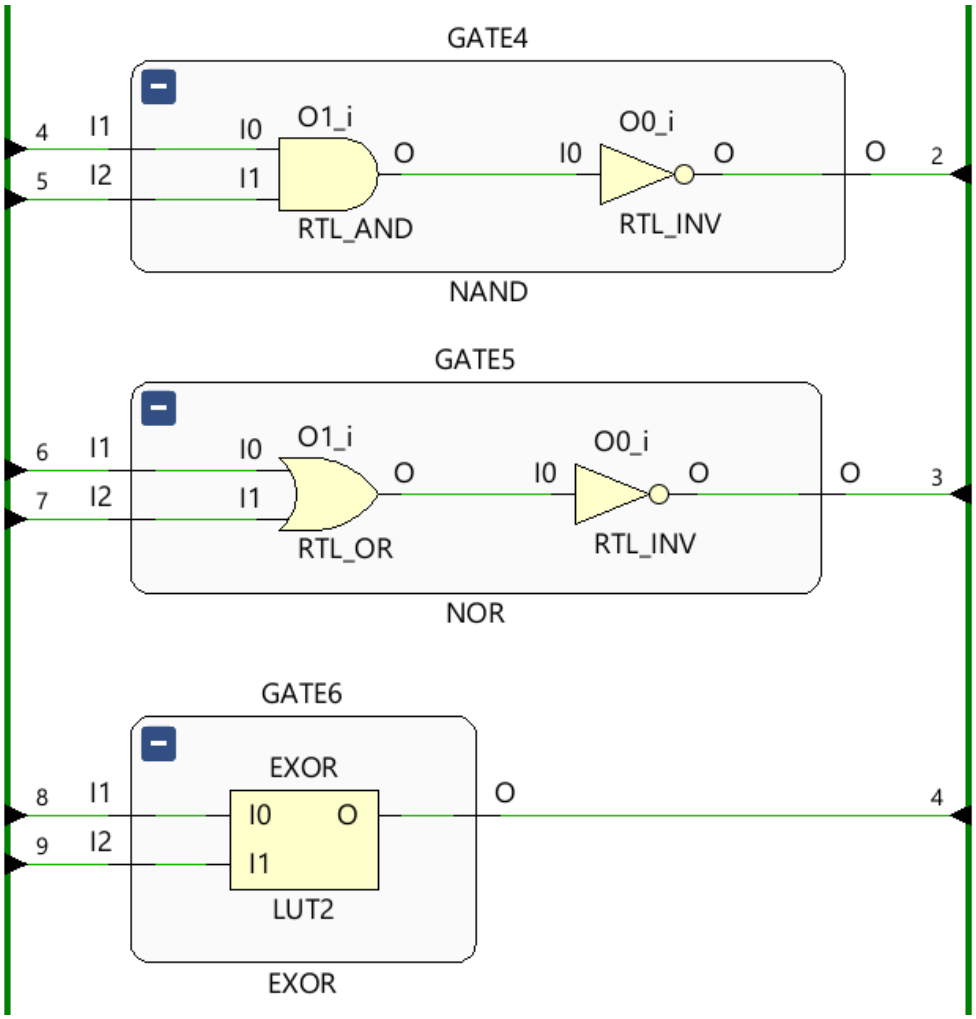
Output 4 exor

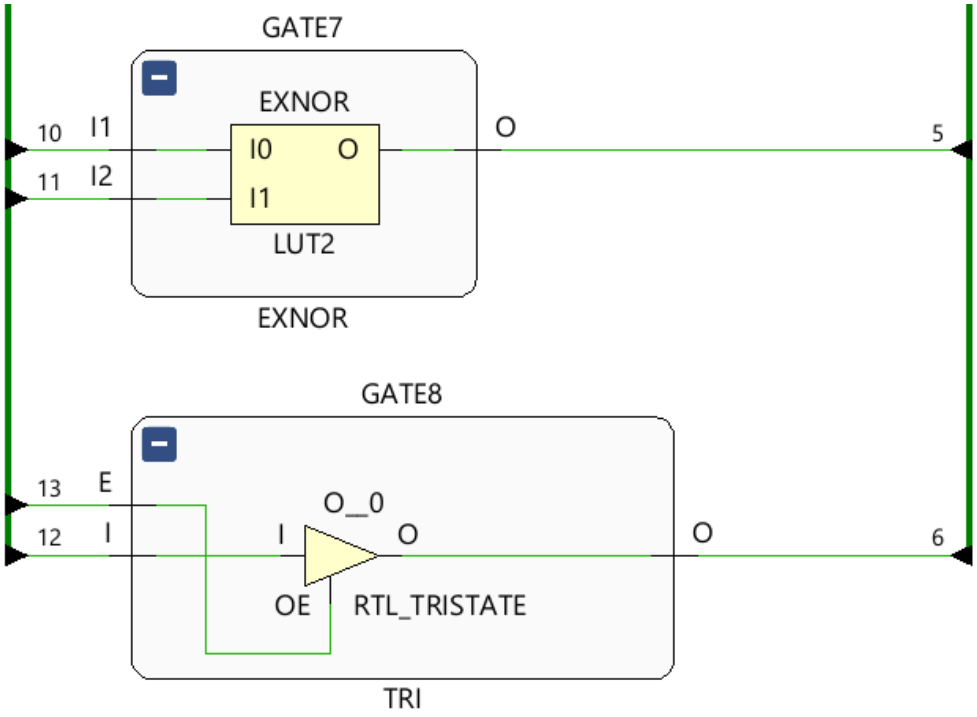
Output 5 exnor

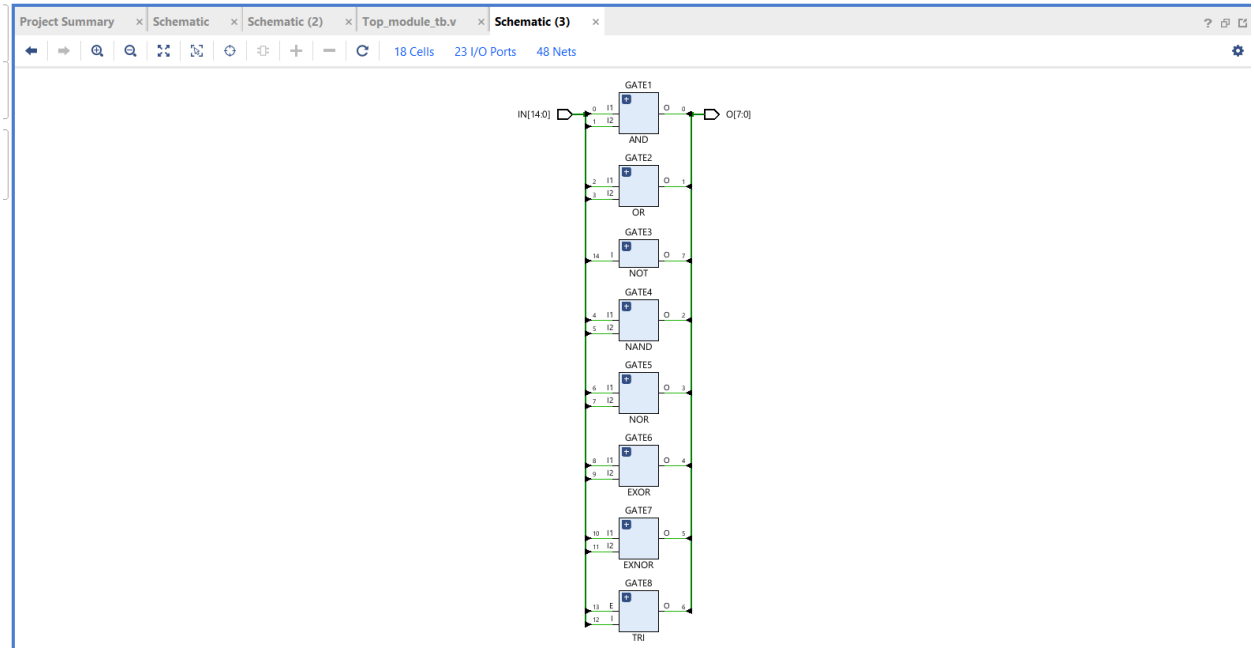
Output 6 tri

Output 7 not //after 20ns I changed the input of the not gate

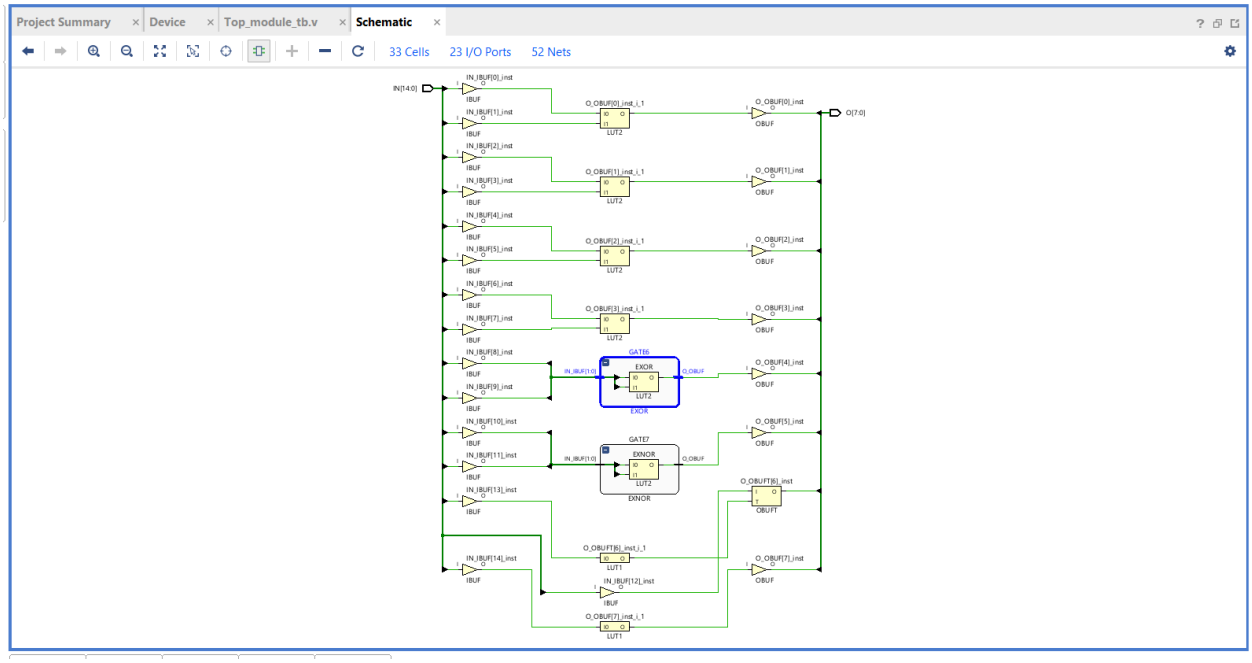

GATE1
IN[14:0]
0
1
I1
I2
I0
I1
O_i
O
O
0
O[7:0]
RTL_AND
AND
GATE2
2
3
I1
I2
I0
I1
O_i
O
O
1
RTL_OR
OR
GATE3
14
I
I0
O_i
O
O
7
RTL_INV
NOT
GATE4


GATE4
4
I1
5
I2
I0
I1
O1_i
O
RTL_AND
I0
O0_i
O
RTL_INV
O
2
NAND
GATE5
6
I1
7
I2
I0
I1
O1_i
O
RTL_OR
I0
O0_i
O
RTL_INV
O
3
NOR
GATE6
EXOR
8
I1
9
I2
I0
I1
O
LUT2
O
4
EXOR


GATE7
EXNOR
10
I1
11
I2
I0
O
I1
LUT2
O
5
EXNOR
GATE8
13
E
12
I
O_0
I
O
O
6
OE
RTL_TRISTATE
TRI


Project Summary
Schematic
Schematic (2)
Top_module_tb.v
Schematic (3)
18 Cells
23 I/O Ports
48 Nets
IN[14:0]
O[7:0]
GATE1
AND
GATE2
OR
GATE3
NOT
GATE4
NAND
GATE5
NOR
GATE6
EXOR
GATE7
EXNOR
GATE8
TRI

TECHNOLOGY SCHEMATIC



AND

Cell Properties

O_OBUF[0]_inst_i_1

I1	I0	O=I0 ...
0	0	0
0	1	0
1	0	0
1	1	1

OR

Cell Properties

O_OBUF[1]_inst_i_1

I1	I0	O=I0 + I1
0	0	0
0	1	1
1	0	1
1	1	1

NAND

Cell Properties

O_OBUF[2]_inst_i_1

I1	I0	O=!I0 + !I1
0	0	1
0	1	1
1	0	1
1	1	0

NOR

Cell Properties

O_OBUF[3]_inst_i_1

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

EXOR

Cell Properties

EXOR

I1	I0	O=I0 & !I1 + !I0 & I1
0	0	0
0	1	1
1	0	1
1	1	0

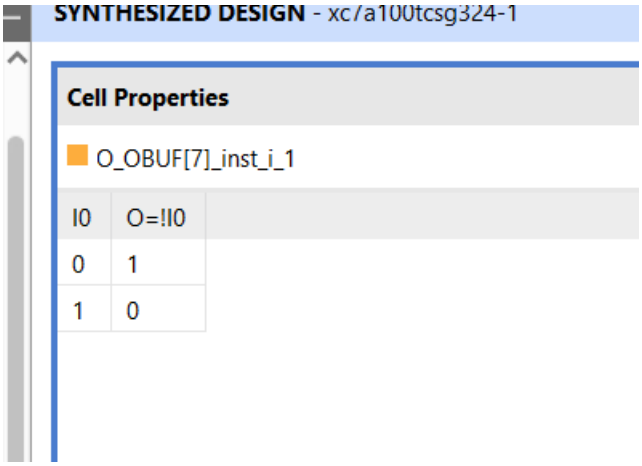
EXNOR

Cell Properties

EXNOR

I1	I0	O=!I0 & !I1 + I0 & I1
0	0	1
0	1	0
1	0	0
1	1	1

NOT


SYNTHESIZED DESIGN - xc7a100tcsg324-1
Cell Properties
O_OBUF[7]_inst_i_1
I0 O=!I0
0 1
1 0

Post-synthesis simulation model

INFO: [VRFC 10-2263] Analyzing Verilog file "C:/Users/omer/Desktop/Dersler/EHB436E/SSTU-01/deney-01/deney-01.sim/sim_1/synth/timing/xsim/top_module_tb_time_synth.v" into library xil_defaultlib

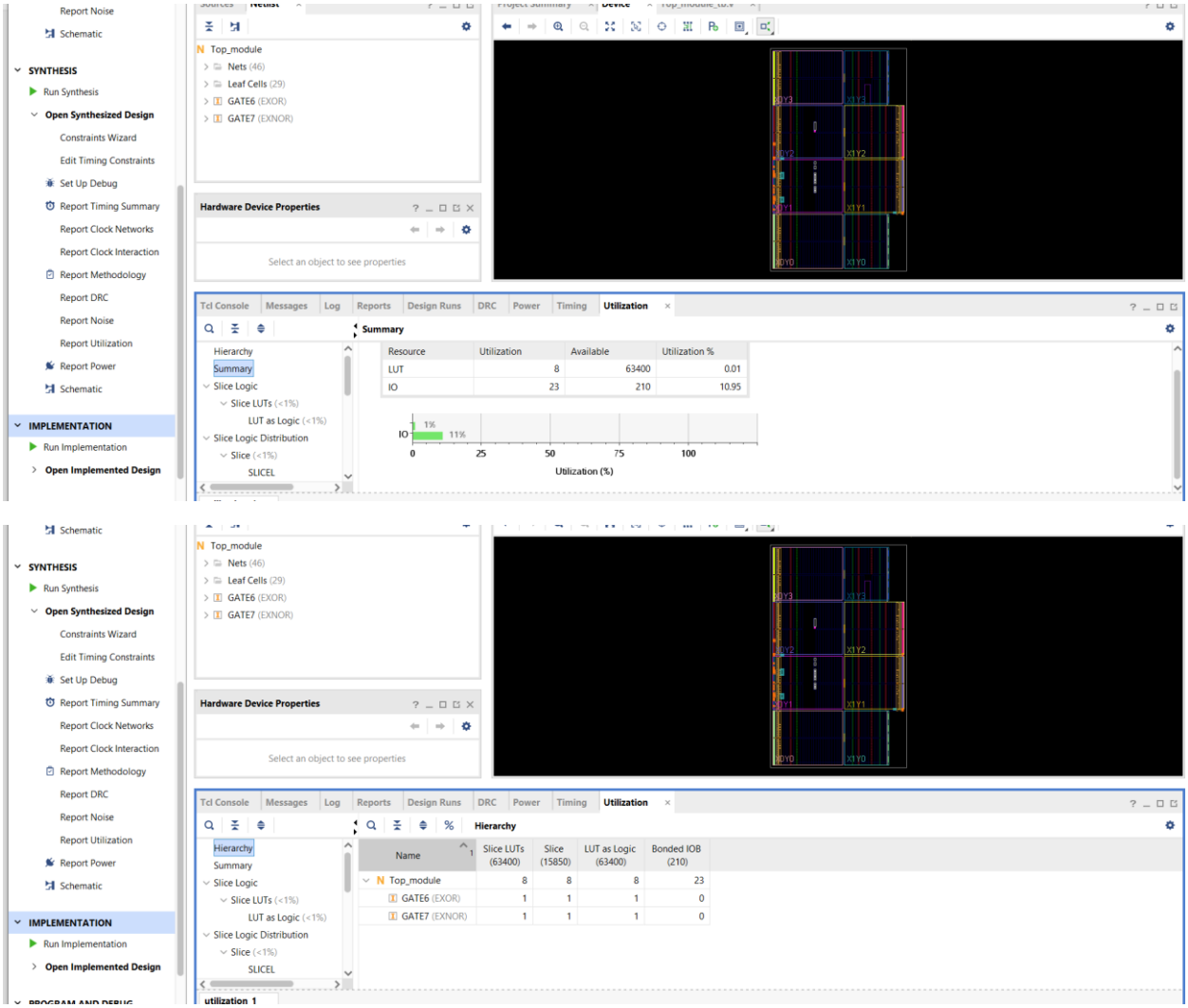
INFO: [VRFC 10-311] analyzing module Top_module

INFO: [VRFC 10-311] analyzing module glbl

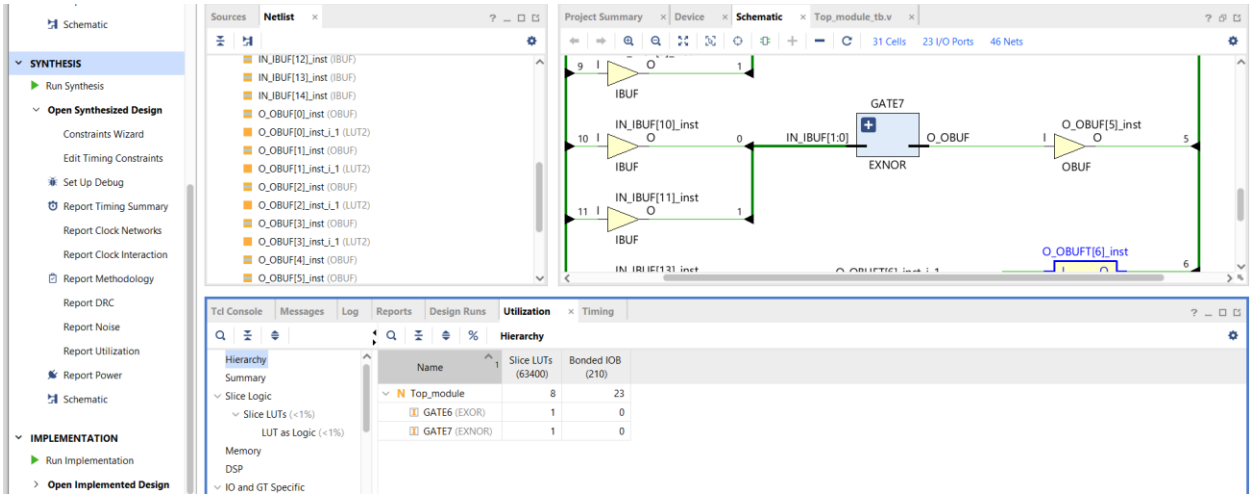
INFO: [VRFC 10-2263] Analyzing Verilog file "C:/Users/omer/Desktop/Dersler/EHB436E/SSTU-01/deney-01/deney-01.srcs/sim_1/new/top_module_tb.v" into library xil_defaultlib

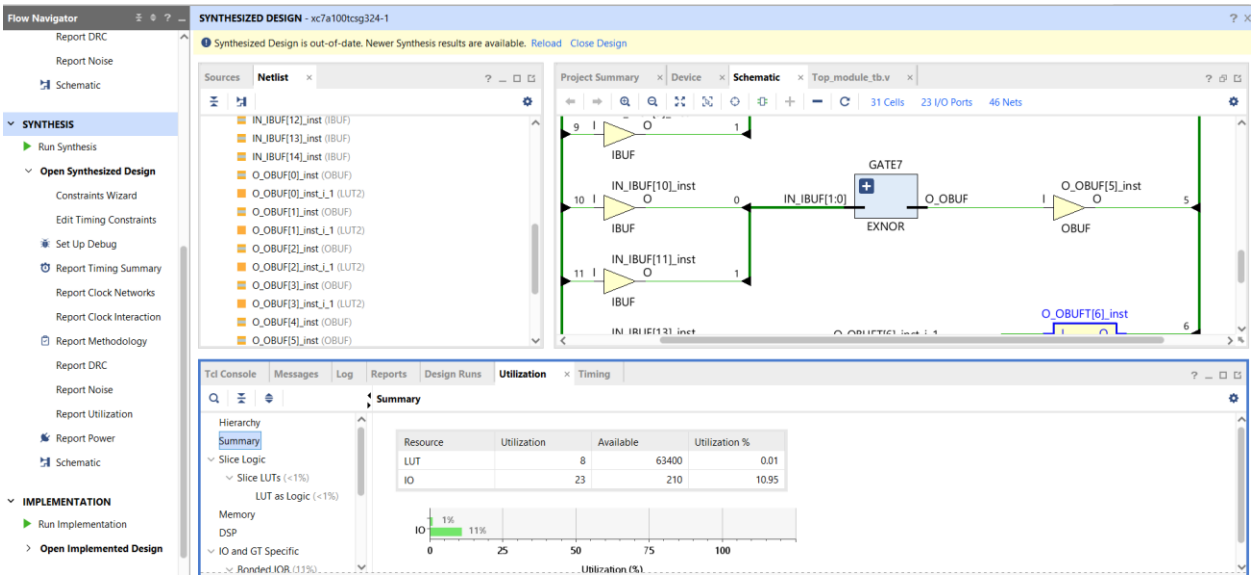
INFO: [VRFC 10-311] analyzing module top_module_tb

UTILIZATION SUMMARY IMPLEMENTATION

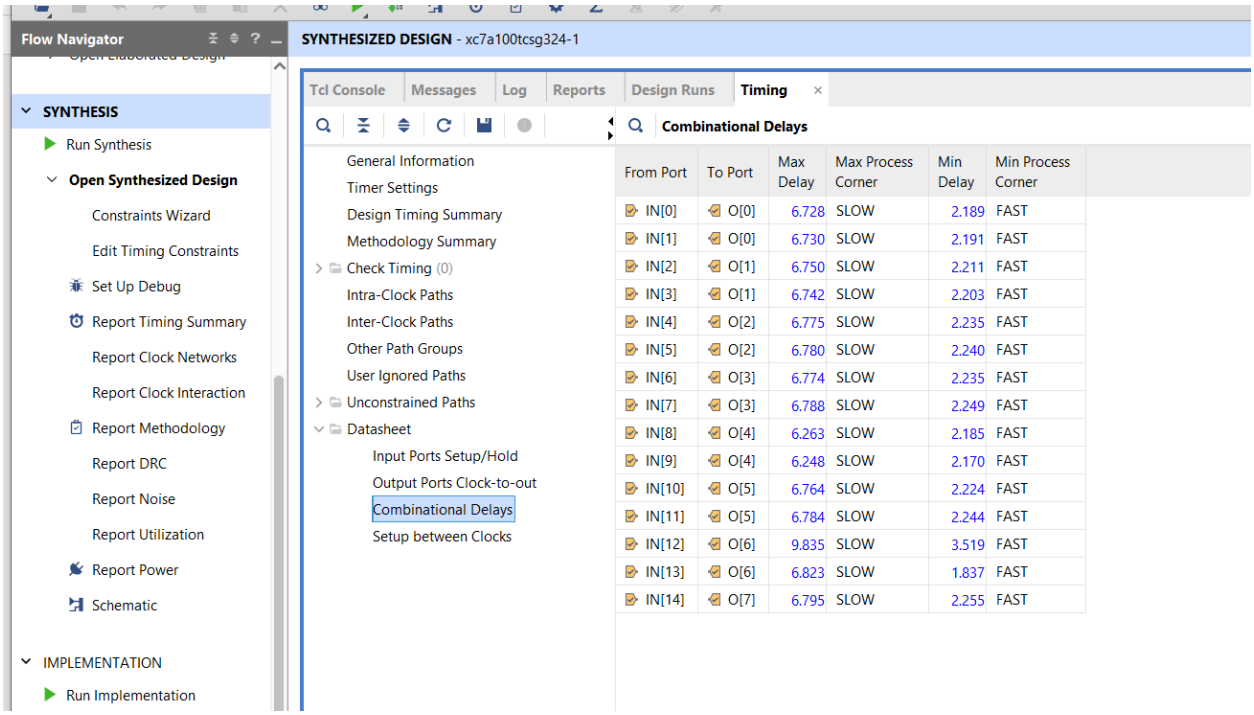


UTILIZATION SUMMARY SYNTHESIS



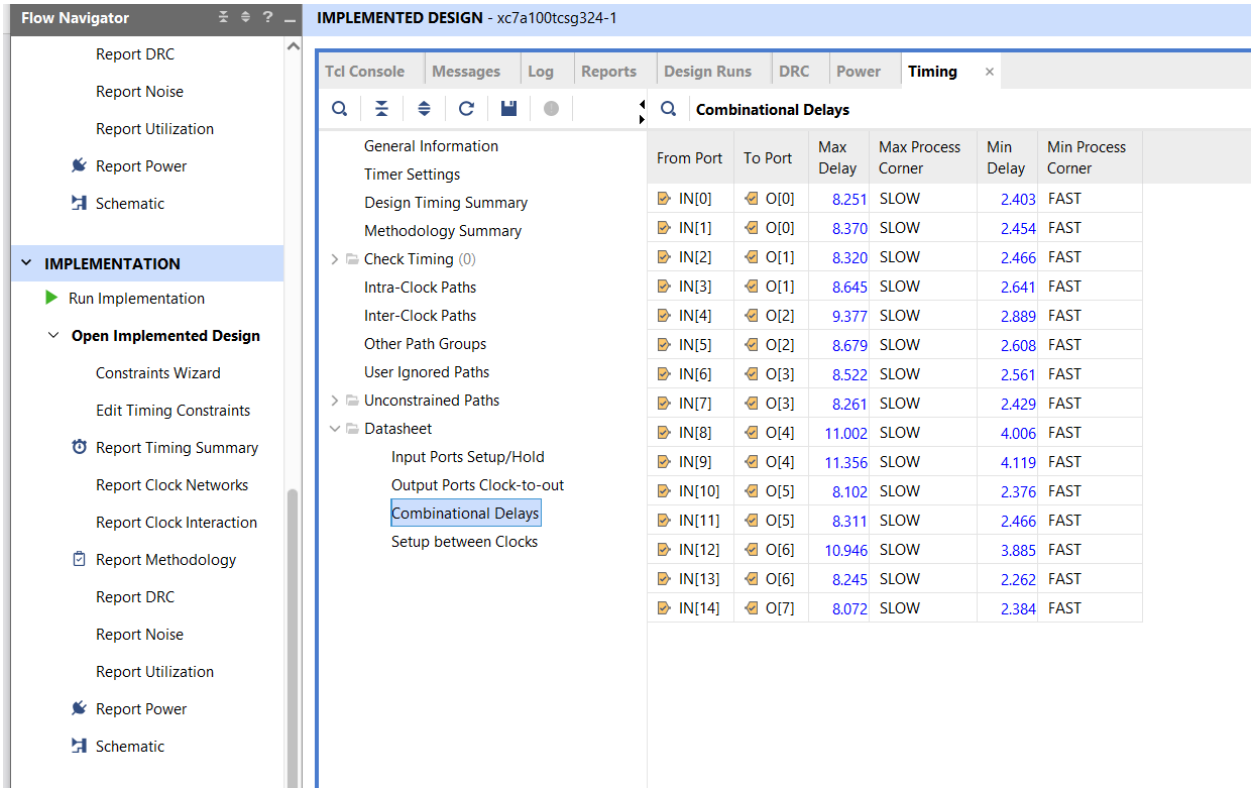


SYNTHESIS TIMING REPORTS



Maximum combinational path delay is 9.835ns

IMPLEMENTATION TIMING REPORT



Maximum combinational path delay is 11.356ns

In synthesis part using ideal components but in implementation part it uses the actual components.

My design does meet the expected results and works properly but maximum combinational path delay is too high.

Fan-in – Fan-out

Fan-in is the maximum input signal that logic cell can be fed and Fan-out is the maximum output signal that logic cell can feed.

LUT

Lookup Table is an array that can store truth tables. Using LUT can be faster compared the computation.

Set-up and Hold-time

Setup-time is the minimum stable input signal time required before the rising edge of the clock for it to be latched

Hold time is the minimum stable input time after clock's active edge.