DIGITAL SYSTEM DESIGN APPLICATION PROJECT 1

İBRAHIM SAMED YALÇIN

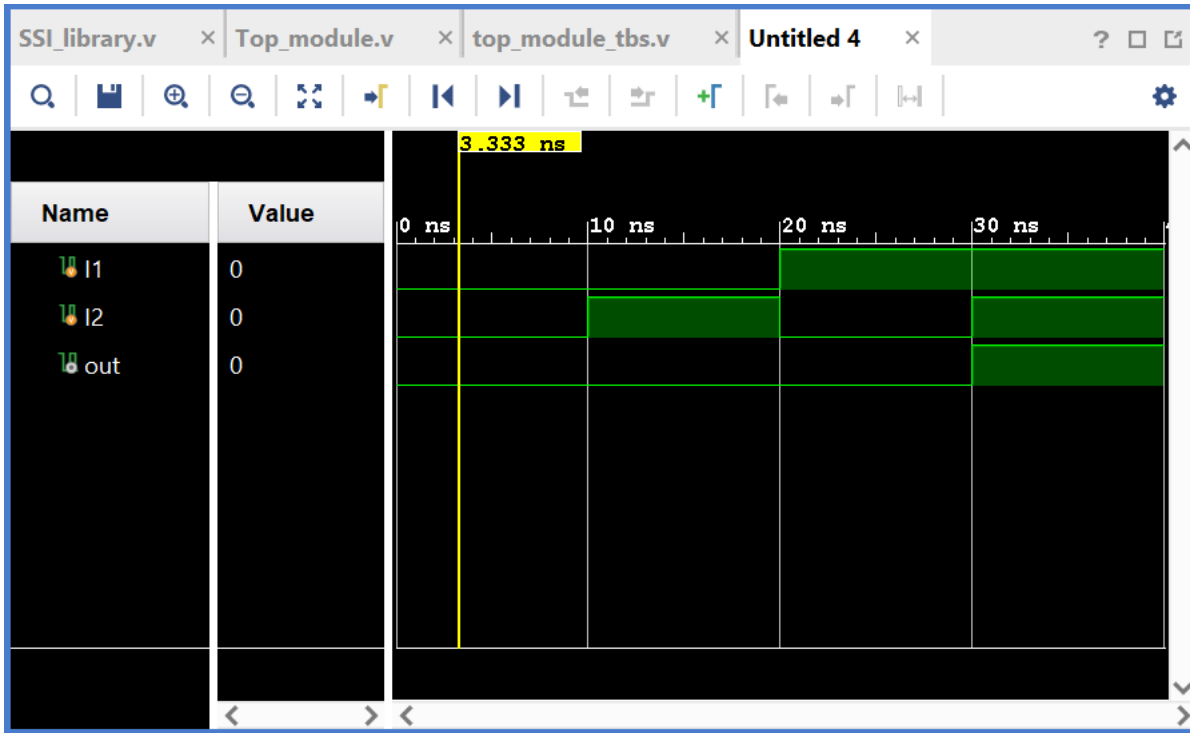
040180055

SSI LIBRARY and GATES

```
`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 I1
    );
    assign O= !I1;
endmodule
module NAND(
    output O,
    input I1,I2
 );
    assign O= !(I1&I2);
endmodule
module EXOR(
    output O,
    input I1,I2
    );
    LUT2 #(
    .INIT ( 4'b1000 )
    ) EXOR
    (
    .I1( I1 ),
    .I2( I2 ),
    .O ( O )
 );
```

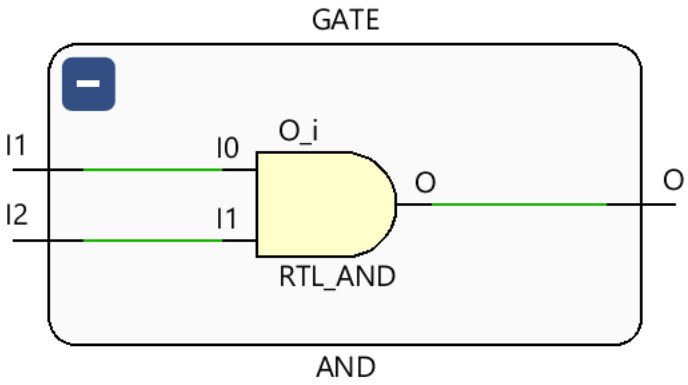
AND GATE

We wrote code for and gate in ssı library.
We created testbench as Top_module_tbs.v
And run simulation -> behavioral simulation, we see that in this table



Our code is working true,

Next, We synthesized design



Truth table

Cell Properties

C_OBUF_inst_i_1

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

Utilization report

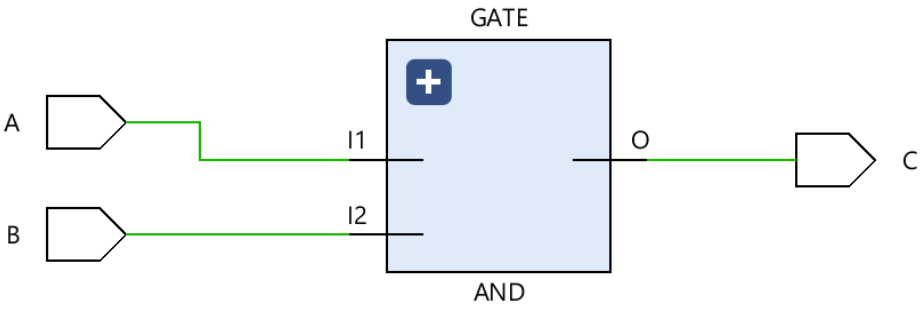
Resource	Utilization	Available	Utilization %
LUT	1	32600	0.00
IO	3	210	1.43

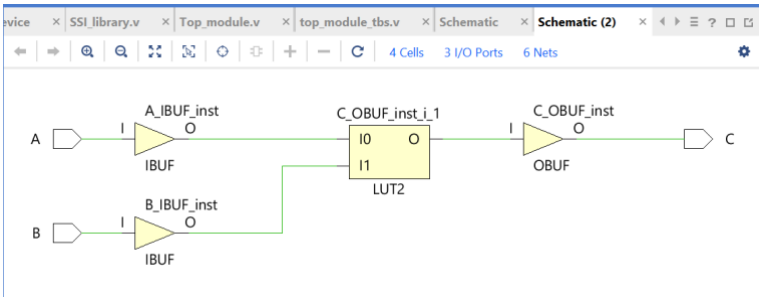
Timing report

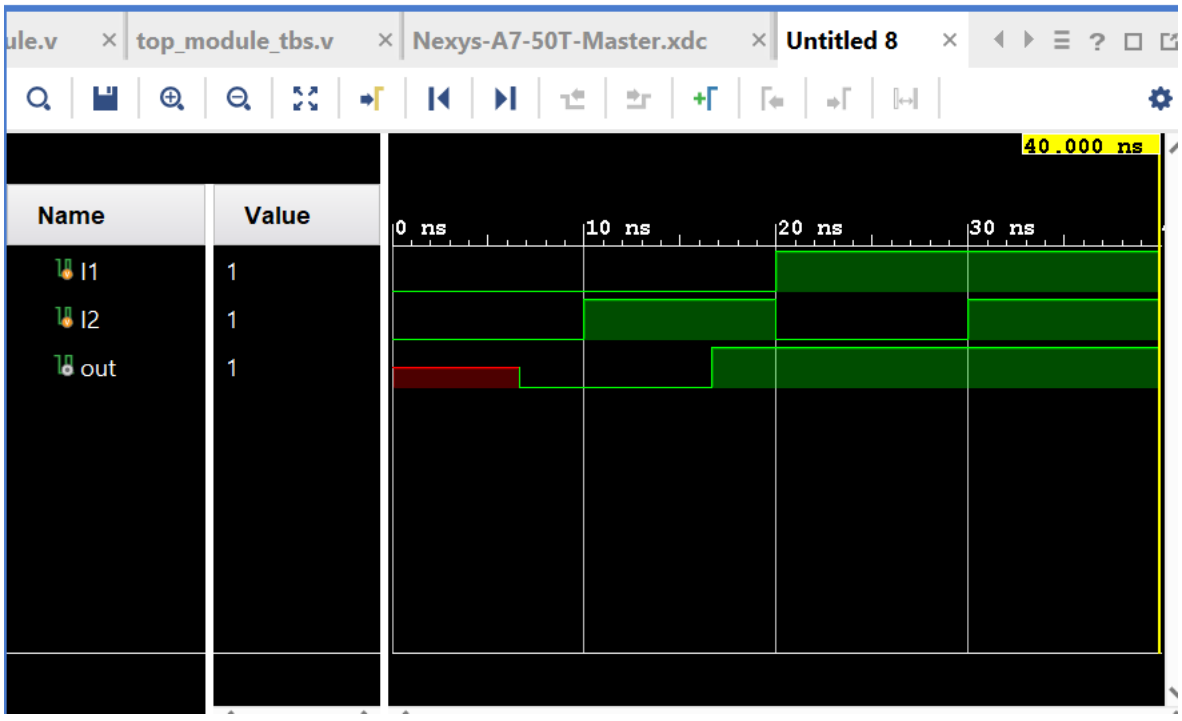
Combinational Delays

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
A	C	6.738	SLOW	2.203	FAST
B	C	6.732	SLOW	2.196	FAST

Rtl and technology schematics

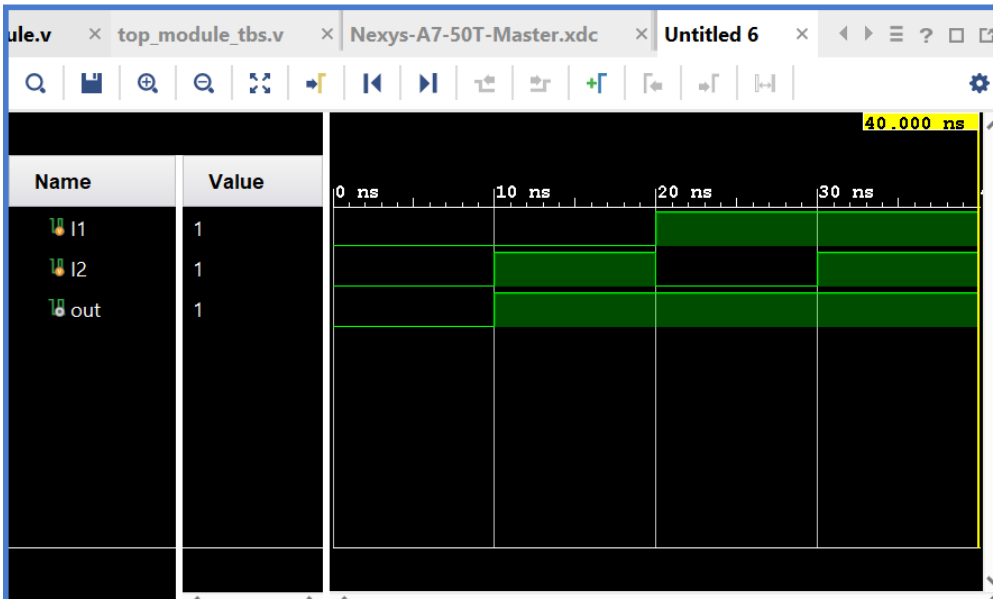


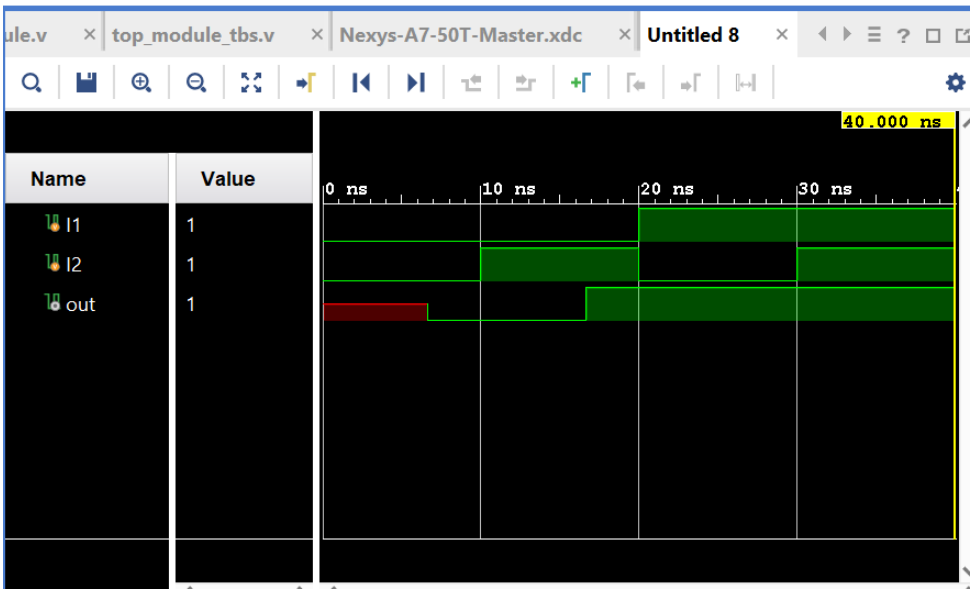


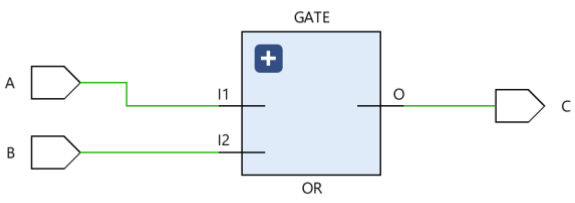


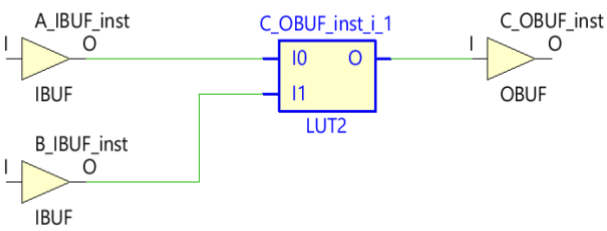
We can see expected behavioral and real behavioral schema are different each other
Cause of this situation is a small time shift occurs while it is realizing. so delay

OR GATE


ule.v
top_module_tbs.v
Nexys-A7-50T-Master.xdc
Untitled 6
40.000 ns
Name
Value
0 ns
10 ns
20 ns
30 ns
I1
1
I2
1
out
1


ule.v
top_module_tbs.v
Nexys-A7-50T-Master.xdc
Untitled 8
40.000 ns
Name
Value
0 ns
10 ns
20 ns
30 ns
I1
1
I2
1
out
1


GATE
A
B
I1
I2
O
C
OR


A_IBUF_inst
I
O
IBUF
C_OBUF_inst_i_1
I0
O
I1
LUT2
C_OBUF_inst
I
O
OBUF
B_IBUF_inst
I
O
IBUF

Other Gates
SSI library

```
`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 I1
);
assign O= !I1;
endmodule
module NAND(
output O,
input I1,I2
);
assign O= !(I1&I2);
endmodule
module NOR(
output O,
input I1,I2
);
assign O=~(I1|I2);
endmodule
module EXOR(
output O,
input I1,I2
);
```

```
module EXNOR(

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

endmodule
```

```
module TRI(

   output O,
```

we already wrote ssi library.
We will create top module. We cannot call of of module like and gate . we can just write their name and ports. We have 15 input port and 8 output port so we define array.

top module

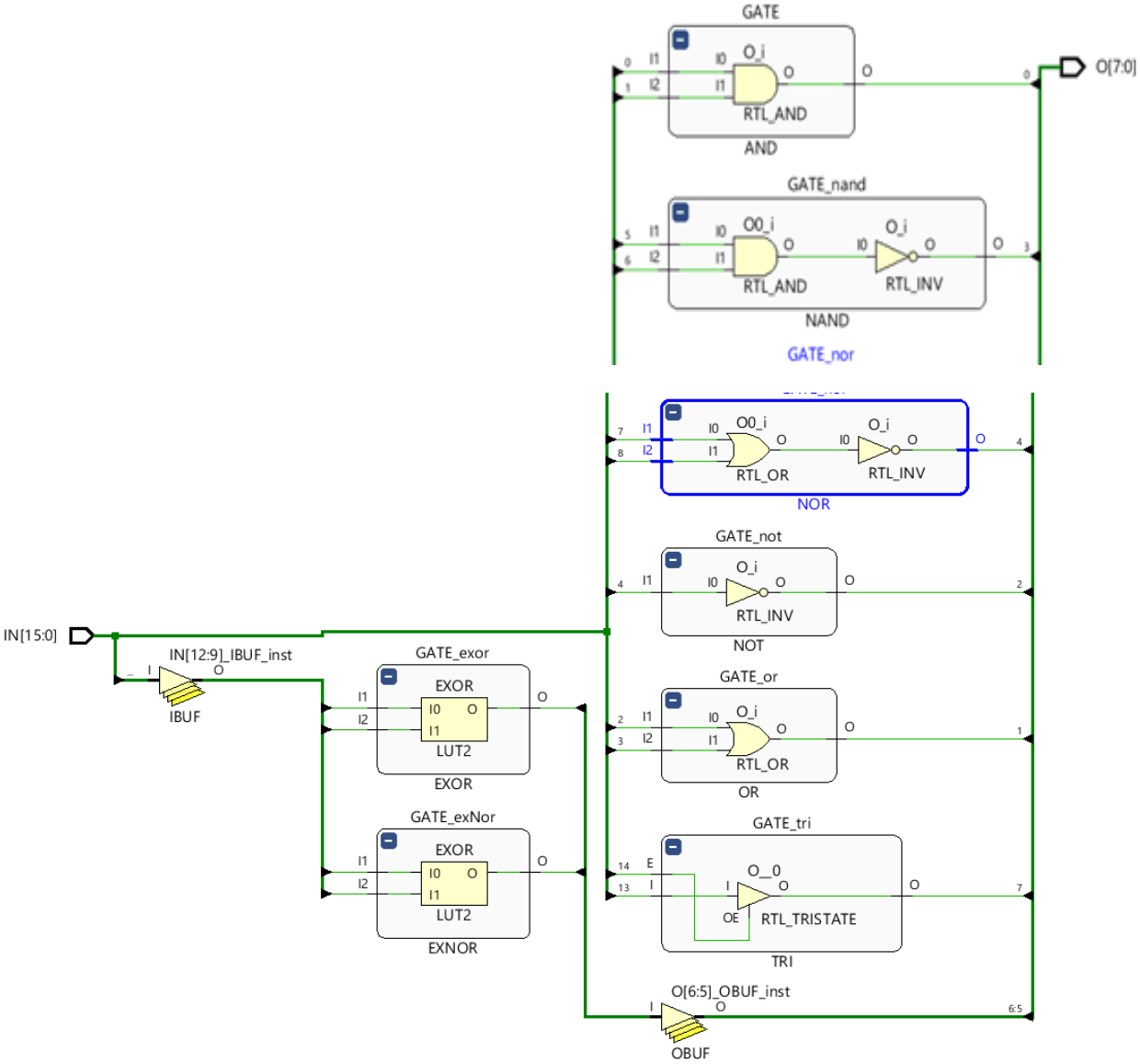
```
`timescale 1ns / 1ps
module Top_module(O,IN);
   output [7:0]O;
   input [15:0]IN;
   AND GATE (.O(O[0]),.I1(IN[0]),.I2(IN[1]));
   OR GATE_or (.O(O[1]),.I1(IN[2]),.I2(IN[3]));
   NOT GATE_not (.O(O[2]),.I1(IN[4]));
   NAND GATE_nand (.O(O[3]),.I1(IN[5]),.I2(IN[6]));
   NOR GATE_nor (.O(O[4]),.I1(IN[7]),.I2(IN[8]));
   EXOR GATE_exor (.O(O[5]),.I1(IN[9]),.I2(IN[10]));
   EXNOR GATE_exNor (.O(O[6]),.I1(IN[11]),.I2(IN[12]));
   TRI GATE_tri (.O(O[7]),.I(IN[13]),.E(IN[14]));
endmodule
```

Now, we create testbench for top module . firstly all of input =0, then 01,10,11
Added 10ms wait time between breaks.

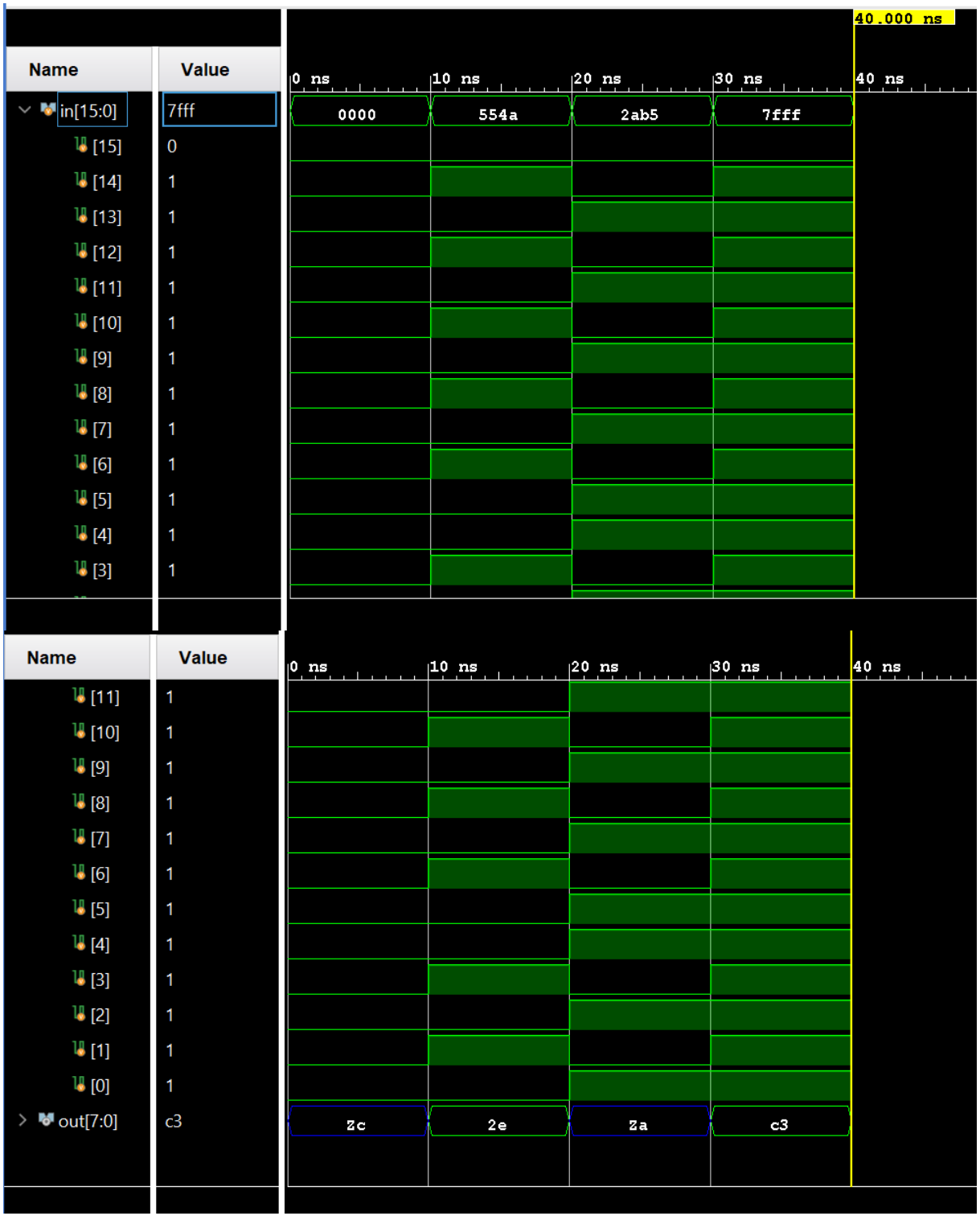
testbench

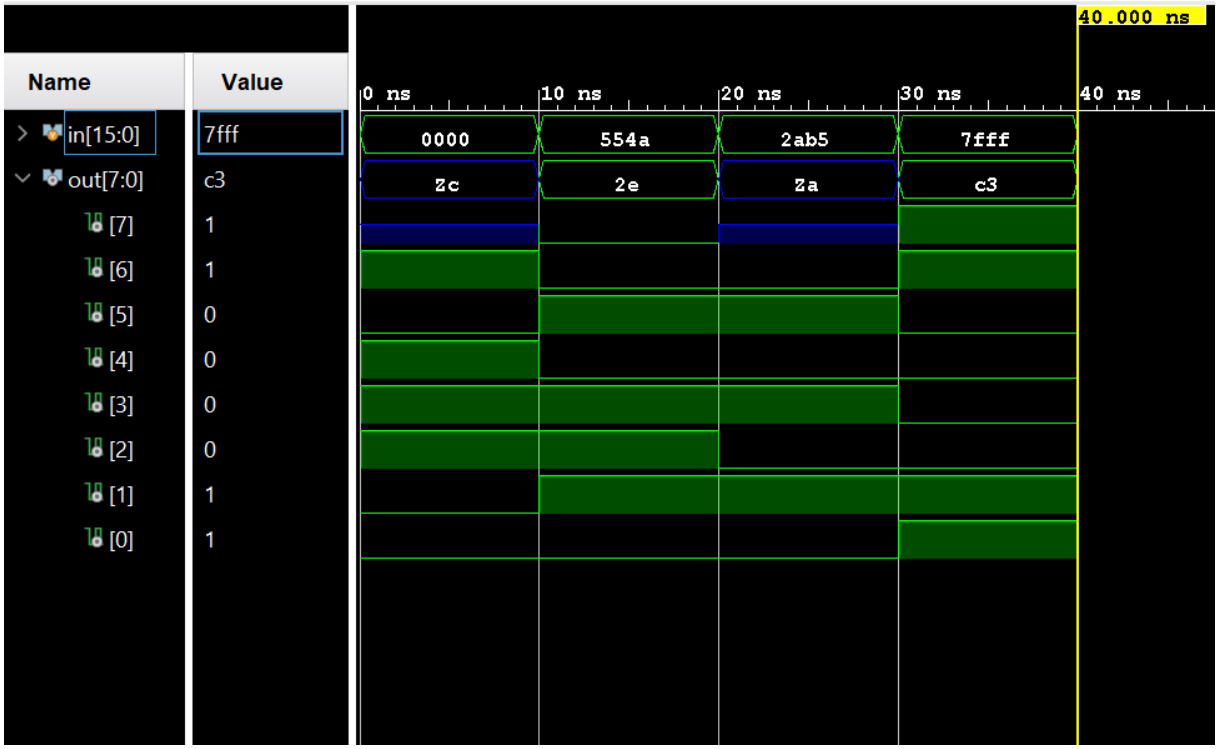
```
`timescale 1ns / 1ps
module top_module_tbs(

    );
    reg [15:0]in;
    wire [7:0]out;

    Top_module top (.O(out),.IN(in));


    initial
    begin

     in=16'h00;
     #10;  in[0]=0; in[1]=1;
     in[2]=0; in[3]=1;
     in[4]=0;
     in[5]=0; in[6]=1;
     in[7]=0; in[8]=1;
     in[9]=0; in[10]=1;
     in[11]=0; in[12]=1;
     in[13]=0; in[14]=1;

     #10;  in[0]=1; in[1]=0;
     in[2]=1; in[3]=0;
     in[4]=1;
     in[5]=1; in[6]=0;
     in[7]=1; in[8]=0;
     in[9]=1; in[10]=0;
     in[11]=1; in[12]=0;
     in[13]=1; in[14]=0;

     #10; in[0]=1; in[1]=1;
     in[2]=1; in[3]=1;
     in[4]=1;
     in[5]=1; in[6]=1;
```

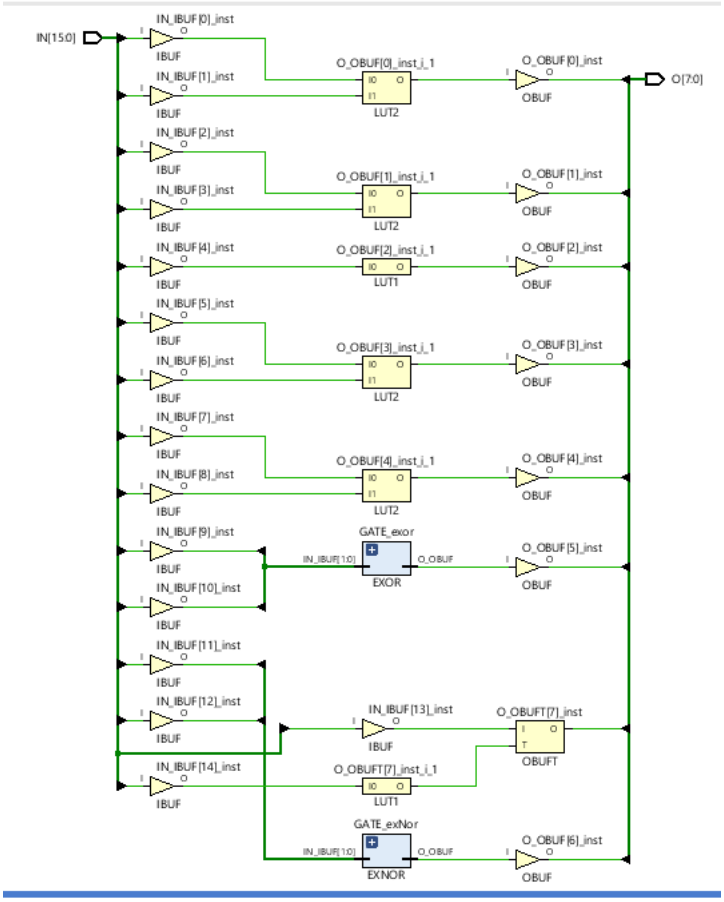
Let's design with the aim of observing the working of the code we wrote. We do that witg elabroted design



This photo are brahvioral shema: 2 photo for input other for output and last photo technological behavioral shema.


40.000 ns
Name
Value
0 ns
10 ns
20 ns
30 ns
40 ns
in[15:0]
7fff
0000
554a
2ab5
7fff
[15] 0
[14] 1
[13] 1
[12] 1
[11] 1
[10] 1
[9] 1
[8] 1
[7] 1
[6] 1
[5] 1
[4] 1
[3] 1
Name
Value
0 ns
10 ns
20 ns
30 ns
40 ns
[11] 1
[10] 1
[9] 1
[8] 1
[7] 1
[6] 1
[5] 1
[4] 1
[3] 1
[2] 1
[1] 1
[0] 1
out[7:0]
c3
zc
2e
za
c3





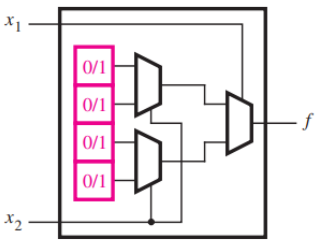
My Project worked on fbga boards. We saw output of all gate via fbga leds. There are not problem.

xvlog.log dosyası

```
INFO: [VRFC 10-2263] Analyzing Verilog file "C:/Users/Ece/Desktop/ibrahim
samed/deneme1/deneme1.sim/sim_1/synth/timing/xsim/top_module_tbs_time_synt
h.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/Ece/Desktop/ibrahim
samed/deneme1/deneme1.srcs/sim_1/new/top_module_tbs.v" into library
xil_defaultlib
INFO: [VRFC 10-311] analyzing module top_module_tbs
```

LUTS

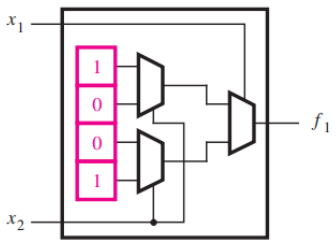
Each logic block in an FPGA typically has a small number of inputs and outputs. A number of FPGA products are on the market, featuring different types of logic blocks. The most commonly used logic block is a lookup table (LUT), which contains storage cells that are used to implement a small logic function. Each cell is capable of holding a single logic value, either 0 or 1. The stored value is produced as the output of the storage cell. LUTs of various sizes may be created, where the size is defined by the number of inputs.



(a) Circuit for a two-input LUT

x_1	x_2	f_1
0	0	1
0	1	0
1	0	0
1	1	1

(b) $f_1 = \bar{x}_1\bar{x}_2 + x_1x_2$



(c) Storage cell contents in the LUT

FAN in FAN out

The fan-in of a logic gate is defined as the number of inputs to the gate. Depending on how a logic gate is constructed, it may be impractical to increase the number of inputs beyond a small number. For example, consider the NMOS NAND gate in Figure 3.53, which has k inputs. We wish to consider the effect of k on the propagation delay tp through the gate. Assume that all k NMOS transistors have the same width W and length L. Because the transistors are connected in series, we can consider them to be equivalent to one long transistor with length k ×L and width W(from book)

SETUP TIME DELAY AND HOLD TIME DELAY

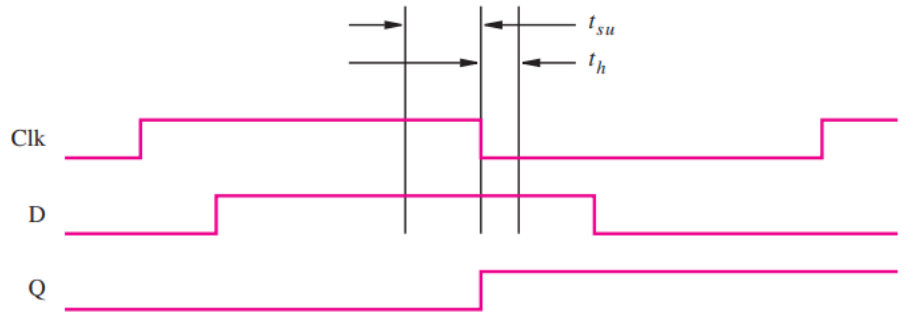


Figure 7.9 Setup and hold times.

Figure illustrates the critical timing region. The minimum time that the D signal must be stable prior to the negative edge of the Clk signal is called the setup time, tsu, of the latch. The minimum time that the D signal must remain stable after the negative edge of the Clk signal is called the hold time, th, of the latch. The values of tsu and th depend on the technology used.

All resource is book that your references book.