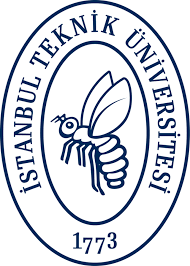
****

**DIGITAL SYSTEM DESIGN APPLICATION – EHB 436E**

**Experiment I**

**Yiğit Bektaş GÜRSOY**

**040180063**

**Class Lecturer: Sıddıka Berna Örs Yalçın**

**Class Assistant:  
Serdar Duran  
Yasin Fırat Kula  
Mehmet Onur Demirtürk**

1. **AND GATE**

* Verilog code,testbench code and behavioral simulation wave screenshots.

**VERILOG CODE**

/\* AND GATE \*/

**module** AND\_gate **(**

**input** I1**,**

**input** I2**,**

**output** O

**);**

**assign** O **=** I1 **&** I2**;**

**endmodule**

**module** Top\_Module\_tb**();**

**reg** **[**1**:**0**]**IN**;**

**wire** **[**0**:**0**]**OUT**;**

Top\_Module DUT**(.**IN**(**IN**),**

**.**OUT**(**OUT**)**

**);**

**initial**

//AND GATE

**begin**

IN**[**0**]=**0**;**

IN**[**1**]=**0**;**

**#**10

IN**[**0**]=**1**;**

IN**[**1**]=**0**;**

**#**10

IN**[**0**]=**0**;**

IN**[**1**]=**1**;**

**#**10

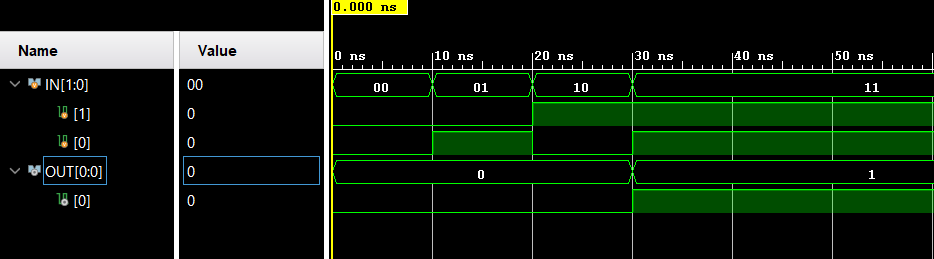
IN**[**0**]=**1**;**

IN**[**1**]=**1**;**

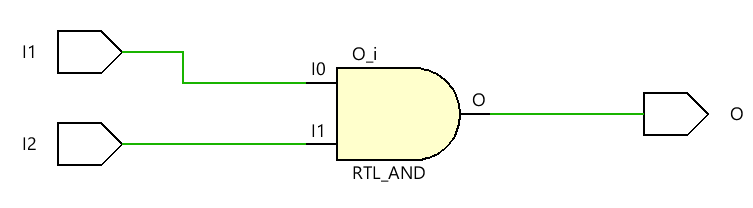
**end**

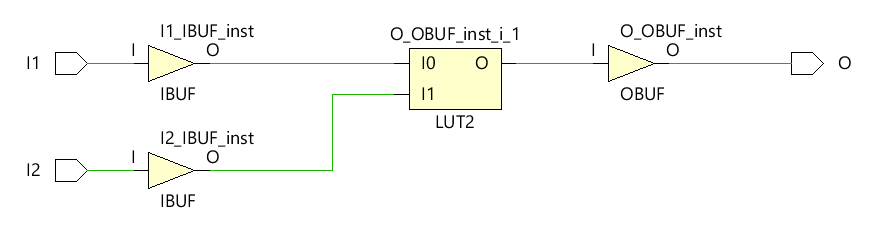
**endmodule**

**BEHAVIORAL SIMULATION**

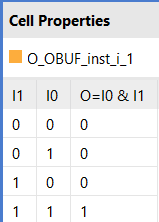
****

As you can see, it outputs 1 only when two inputs are "1", and outputs 0 when one of the 2 inputs is 0. As a result of this simulation, we can see that we have designed and door correctly.

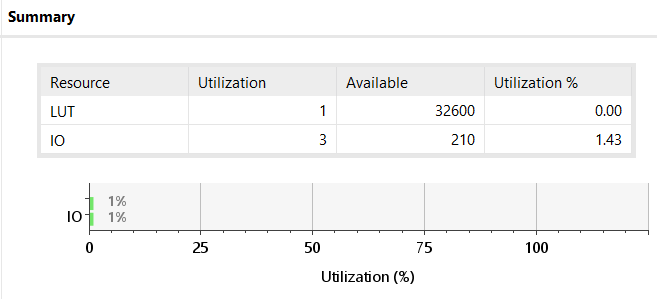
* **RTL Schematic**
* **Technology Schematic**

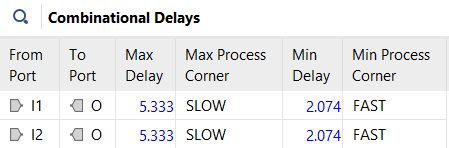


* The accuracy of the above technology and RTL schematics can be checked with the following truth table. When checked, it seems that they both give the same output and the output values of our schematics overlap with each other.
* **Synthesis Report**
* Truth Table of the LUT



* Usage of the FPGA resources (utilization summary)



* Combinational path delays
* Maximum combinational path delay is 5.3333.
* Post-synthesis simulation model

//-------- JTAG Globals --------------

**wire** JTAG\_TDO\_GLBL**;**

**wire** JTAG\_TCK\_GLBL**;**

**wire** JTAG\_TDI\_GLBL**;**

**wire** JTAG\_TMS\_GLBL**;**

**wire** JTAG\_TRST\_GLBL**;**

**reg** JTAG\_CAPTURE\_GLBL**;**

**reg** JTAG\_RESET\_GLBL**;**

**reg** JTAG\_SHIFT\_GLBL**;**

**reg** JTAG\_UPDATE\_GLBL**;**

**reg** JTAG\_RUNTEST\_GLBL**;**

**reg** JTAG\_SEL1\_GLBL **=** 0**;**

**reg** JTAG\_SEL2\_GLBL **=** 0 **;**

**reg** JTAG\_SEL3\_GLBL **=** 0**;**

**reg** JTAG\_SEL4\_GLBL **=** 0**;**

**reg** JTAG\_USER\_TDO1\_GLBL **=** 1'bz**;**

**reg** JTAG\_USER\_TDO2\_GLBL **=** 1'bz**;**

**reg** JTAG\_USER\_TDO3\_GLBL **=** 1'bz**;**

**reg** JTAG\_USER\_TDO4\_GLBL **=** 1'bz**;**

**assign** **(strong1,** **weak0)** GSR **=** GSR\_int**;**

**assign** **(strong1,** **weak0)** GTS **=** GTS\_int**;**

**assign** **(weak1,** **weak0)** PRLD **=** PRLD\_int**;**

**initial** **begin**

GSR\_int **=** 1'b1**;**

PRLD\_int **=** 1'b1**;**

**#(**ROC\_WIDTH**)**

GSR\_int **=** 1'b0**;**

PRLD\_int **=** 1'b0**;**

**end**

**initial** **begin**

GTS\_int **=** 1'b1**;**

**#(**TOC\_WIDTH**)**

GTS\_int **=** 1'b0**;**

**end**

**endmodule**

`endif

`timescale 1 ps **/** 1 ps

`define XIL\_TIMING

**(\*** NotValidForBitStream **\*)**

**module** AND\_gate

**(**I1**,**

I2**,**

O**);**

**input** I1**;**

**input** I2**;**

**output** O**;**

**wire** I1**;**

**wire** I1\_IBUF**;**

**wire** I2**;**

**wire** I2\_IBUF**;**

**wire** O**;**

**wire** O\_OBUF**;**

**initial** **begin**

$sdf\_annotate**(**"Top\_Module\_tb\_time\_synth.sdf"**,,,,**"tool\_control"**);**

**end**

IBUF I1\_IBUF\_inst

**(.**I**(**I1**),**

**.**O**(**I1\_IBUF**));**

IBUF I2\_IBUF\_inst

**(.**I**(**I2**),**

**.**O**(**I2\_IBUF**));**

OBUF O\_OBUF\_inst

**(.**I**(**O\_OBUF**),**

**.**O**(**O**));**

LUT2 **#(**

**.**INIT**(**4'h8**))**

O\_OBUF\_inst\_i\_1

**(.**I0**(**I1\_IBUF**),**

**.**I1**(**I2\_IBUF**),**

**.**O**(**O\_OBUF**));**

**endmodule**

`ifndef GLBL

`define GLBL

`timescale 1 ps **/** 1 ps

**module** glbl **();**

**parameter** ROC\_WIDTH **=** 100000**;**

**parameter** TOC\_WIDTH **=** 0**;**

//-------- STARTUP Globals --------------

**wire** GSR**;**

**wire** GTS**;**

**wire** GWE**;**

**wire** PRLD**;**

**tri1** p\_up\_tmp**;**

**tri** **(weak1,** **strong0)** PLL\_LOCKG **=** p\_up\_tmp**;**

**wire** PROGB\_GLBL**;**

**wire** CCLKO\_GLBL**;**

**wire** FCSBO\_GLBL**;**

**wire** **[**3**:**0**]** DO\_GLBL**;**

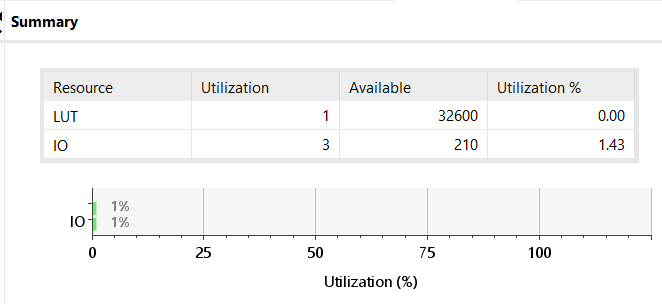
**wire** **[**3**:**0**]** DI\_GLBL**;**

**reg** GSR\_int**;**

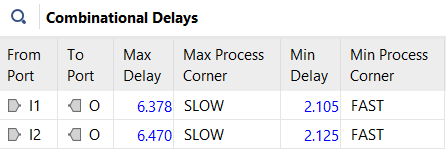
**reg** GTS\_int**;**

**reg** PRLD\_int**;**

* **Implementation Report:**
* Usage of the FPGA resources (utilization summary)



* Combinational path delays



* Maximum combinational path delay is 6470ns.
* Implementation result was higher than synthesis result. The reason for this is that the circuit that we get output from as a result of the implementation is our circuit that makes the mapping on the FPGA. When we synthesize, the components are not processed on the FPGA. In this direction, the implementation results give more realistic results in delay times.

1. **OTHER GATES**

* Verilog code,testbench code and behavioral simulation wave screenshots.

**VERILOG CODES**

**module** EXOR\_gate**(**

**input** I1**,**

**input** I2**,**

**output** O

**);**

LUT2 **#(**

**.**INIT **(** 4'b0110 **)**

**)** EXOR

**(**

**.**I0**(** I1 **),**

**.**I1**(** I2 **),**

**.**O **(** O **)**

**);**

**endmodule**

**module** EXNOR\_gate**(**

**input** I1**,**

**input** I2**,**

**output** O

**);**

LUT2 **#(**

**.**INIT **(** 4'b1001 **)**

**)** EXOR

**(**

**.**I0**(** I1 **),**

**.**I1**(** I2 **),**

**.**O **(** O **)**

**);**

**endmodule**

**module** TRI**(**

**input** I**,**

**input** E**,**

**output** O

**);**

**assign** O **=** **(** E **==** 1'b1**)** **?** I **:** 1'bZ**;**

**endmodule**

/\* OR GATE \*/

**module** OR\_gate **(**

**input** I1**,**

**input** I2**,**

**output** O

**);**

**assign** O **=** I1 **|** I2**;**

**endmodule**

/\* NOT GATE \*/

**module** NOT\_gate **(**

**input** I**,**

**output** O

**);**

**assign** O **=** **~**I**;**

**endmodule**

**module** NAND\_gate **(**

**input** I1**,**

**input** I2**,**

**output** **reg** O

**);**

**always@\***

**begin**

**assign** O **=** **~(** I1 **&** I2**);**

**end**

**endmodule**

/\* NOR gate \*/

**module** NOR\_gate **(**

**input** I1**,**

**input** I2**,**

**output** **reg** O

**);**

**always@\***

**begin**

**assign** O **=** **~(** I1 **|** I2**);**

**end**

**endmodule**

**TOP MODULE**

NOR\_gate U5**(**

**.**I1**(**IN**[**7**]),**

**.**I2**(**IN**[**8**]),**

**.**O**(**OUT**[**4**]));**

EXOR\_gate U6**(**

**.**I1**(**IN**[**9**]),**

**.**I2**(**IN**[**10**]),**

**.**O**(**OUT**[**5**]));**

EXNOR\_gate U7**(**

**.**I1**(**IN**[**11**]),**

**.**I2**(**IN**[**12**]),**

**.**O**(**OUT**[**6**]));**

TRI U8**(**

**.**I**(**IN**[**13**]),**

**.**E**(**IN**[**14**]),**

**.**O**(**OUT**[**7**]));**

**endmodule**

**module** Top\_Module**(**

**input** **[**14**:**0**]**IN**,**

**output** **[**7**:**0**]**OUT

**);**

AND\_gate U1**(**

**.**I1**(**IN**[**0**]),**

**.**I2**(**IN**[**1**]),**

**.**O**(**OUT**[**0**]));**

OR\_gate U2**(**

**.**I1**(**IN**[**2**]),**

**.**I2**(**IN**[**3**]),**

**.**O**(**OUT**[**1**]));**

NOT\_gate U3**(**

**.**I**(**IN**[**4**]),**

**.**O**(**OUT**[**2**]));**

NAND\_gate U4**(**

**.**I1**(**IN**[**5**]),**

**.**I2**(**IN**[**6**]),**

**.**O**(**OUT**[**3**]));**

**TOP MODULE TEST BENCH**

//NOR GATE

**#**10

IN**[**7**]=**0**;**IN**[**8**]=**0**;**

**#**10

IN**[**7**]=**1**;**IN**[**8**]=**0**;**

**#**10

IN**[**7**]=**0**;**IN**[**8**]=**1**;**

**#**10

IN**[**7**]=**1**;**IN**[**8**]=**1**;**

//EXOR GATE

**#**10

IN**[**9**]=**0**;**IN**[**10**]=**0**;**

**#**10

IN**[**9**]=**1**;**IN**[**10**]=**0**;**

**#**10

IN**[**9**]=**0**;**IN**[**10**]=**1**;**

**#**10

IN**[**9**]=**1**;**IN**[**10**]=**1**;**

//EXNOR GATE

**#**10

IN**[**11**]=**0**;**IN**[**12**]=**0**;**

**#**10

IN**[**11**]=**1**;**IN**[**12**]=**0**;**

**#**10

IN**[**11**]=**0**;**IN**[**12**]=**1**;**

**#**10

IN**[**11**]=**1**;**IN**[**12**]=**1**;**

//TRI GATE

**#**10

IN**[**13**]=**0**;**IN**[**14**]=**0**;**

**#**10

IN**[**13**]=**1**;**IN**[**14**]=**0**;**

**#**10

IN**[**13**]=**0**;**IN**[**14**]=**1**;**

**#**10

IN**[**13**]=**1**;**IN**[**14**]=**1**;**

**end**

**endmodule**

timescale 1ns **/** 1ps

**module** Top\_Module\_tb**();**

**reg** **[**14**:**0**]**IN**;**

**wire** **[**7**:**0**]**OUT**;**

Top\_Module DUT**(.**IN**(**IN**),**

**.**OUT**(**OUT**)**

**);**

**initial**

//AND GATE

**begin**

IN**[**0**]=**0**;**IN**[**1**]=**0**;**

**#**10

IN**[**0**]=**1**;**IN**[**1**]=**0**;**

**#**10

IN**[**0**]=**0**;**IN**[**1**]=**1**;**

**#**10

IN**[**0**]=**1**;**IN**[**1**]=**1**;**

//OR GATE

**#**10

IN**[**2**]=**0**;**IN**[**3**]=**0**;**

**#**10

IN**[**2**]=**1**;**IN**[**3**]=**0**;**

**#**10

IN**[**2**]=**0**;**IN**[**3**]=**1**;**

**#**10

IN**[**2**]=**1**;**IN**[**3**]=**1**;**

//NOT GATE

**#**10

IN**[**4**]=**0**;**

**#**10

IN**[**4**]=**1**;**

//NAND GATE

**#**10

IN**[**5**]=**0**;**IN**[**6**]=**0**;**

**#**10

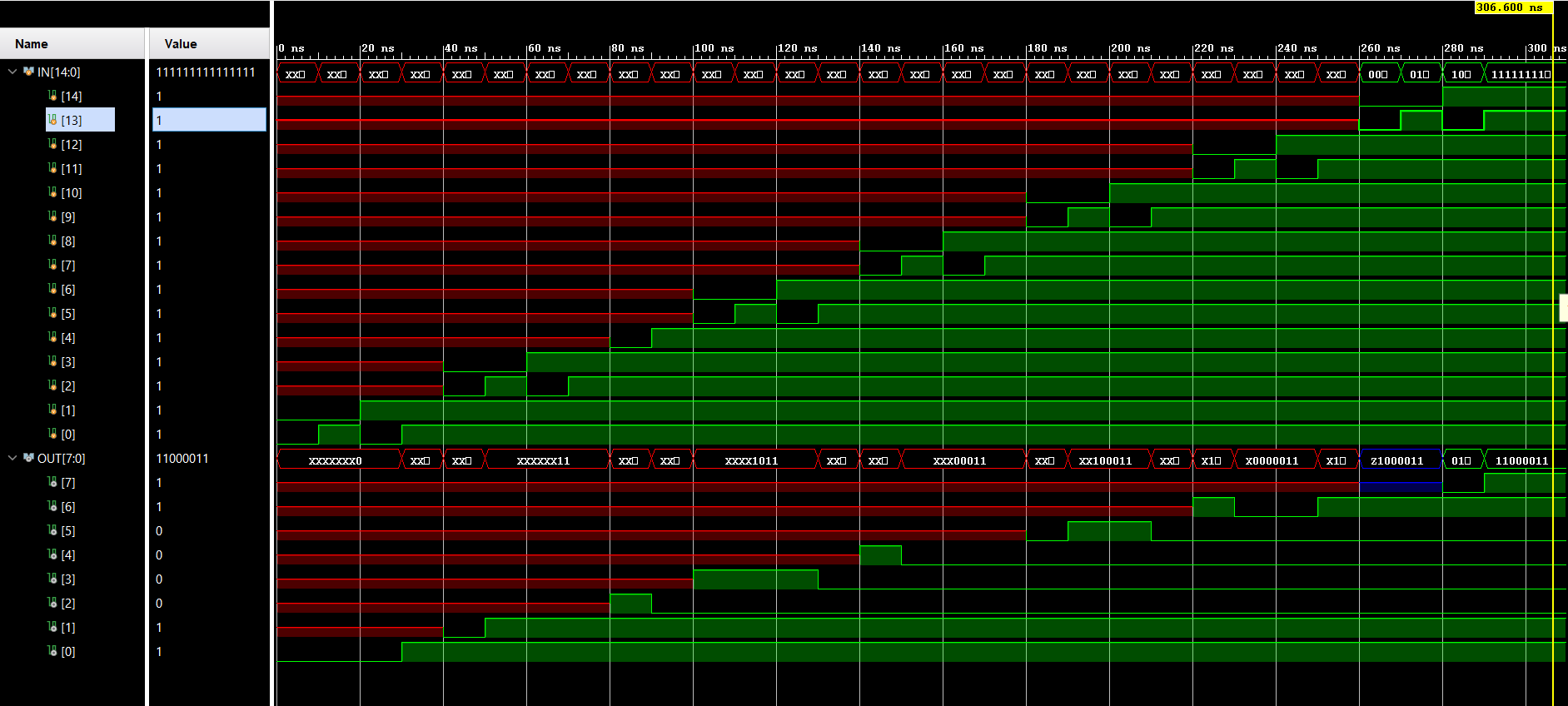
IN**[**5**]=**1**;**IN**[**6**]=**0**;**

**#**10

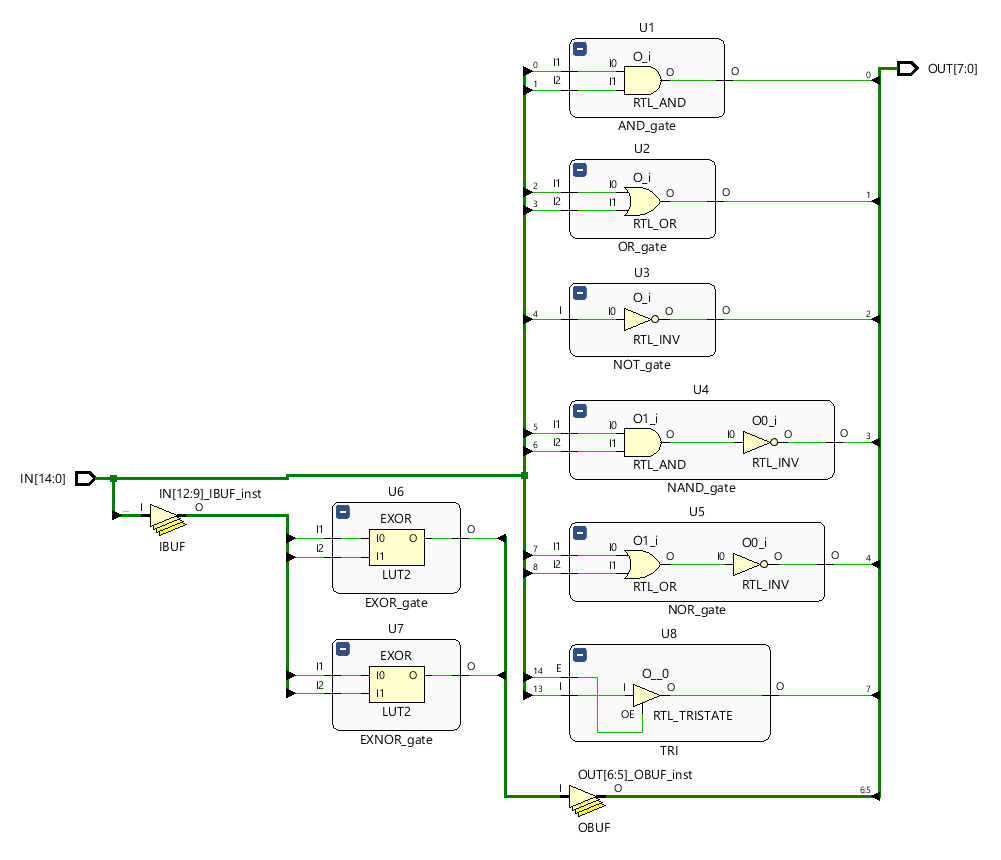
IN**[**5**]=**0**;**IN**[**6**]=**1**;**

**#**10

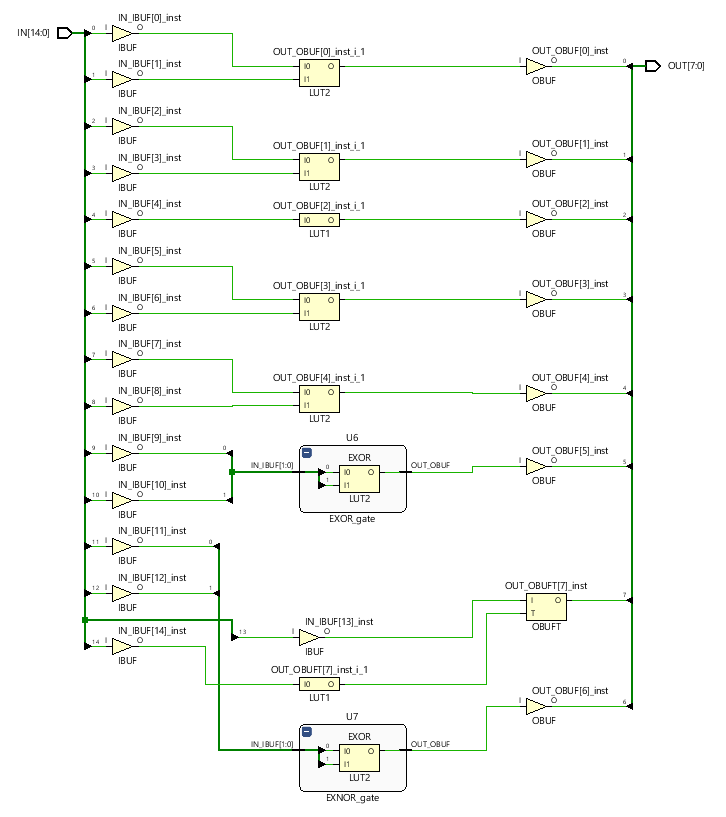
IN**[**5**]=**1**;**IN**[**6**]=**1**;**

**BEHAVIORAL SIMULATION**

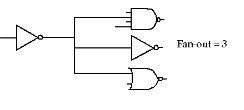
* As seen in the simulation results, all of our gates give the correct result. All doors have been tested with 10 second intervals.
* **RTL Schmematic**



* **Technology schematic**



* **Look-up Table (LUT):**
* Lookup table is a table that generates data based on input values. It is frequently used in simulation programs. It is an array that holds values that need to be calculated. The lookup table can be filled in manually when the program is written, or the simulation program can fill it with values while calculating the lookup table. When values are needed later, the program can save CPU resources by searching for them.
* **Fan-in and fan-out:**
* Fan-in: Fan input is a term used to describe the maximum number of inputs to a logic gate (logic circuit). For example, let's have a circuit like the image below. Suppose we have a 3-input NAND gate. Based on what has been said, the fan-in value is 3.



* Fan-out: The maximum number of loads that a logic gate can drive from the same integrated family is called , Fan Out, the output capacity. The maximum fan out of an output measures the load-driving capacity.
* **Setup time and hold time:**
* Setup time: Setup time is defined as the minimum time before the active edge of the clock that must be constant for data to be retained correctly. For a flip-flop or any sequential item to hold data reliably, it needs time to pass before the clock edge arrives for the data to remain constant. This time is known as the setup time.
* Hold time: Retention time is defined as the minimum time after the active edge of the clock during which data must be stable. The data needs some time to remain stable after the clock edge comes to capture data losslessly. This is the holding time.

