

JK LAKSHMIPAT UNIVERSITY

DIGITAL CIRCUIT AND SYSTEMS  
(EE1120)

Activity 04

Logic gates using VHDL

Date : 16th February 2023

Name : Bobby Sharma

AIM: Design all the logic gates using VHDL (Hardware Description Language).

APPARATUS REQUIRED: VHDL in your device.

THEORY:

VHDL stands for very high-speed integrated circuit hardware description language. It is a programming language used to model a digital system by dataflow, behavioral and structural style of modeling. This language was first introduced in 1981 for the department of Defense (DoD) under the VHSIC program.

A logic gate is a simple switching circuit that determines whether an input pulse can pass through to the output in digital circuits. The following types of logic gates are commonly used:

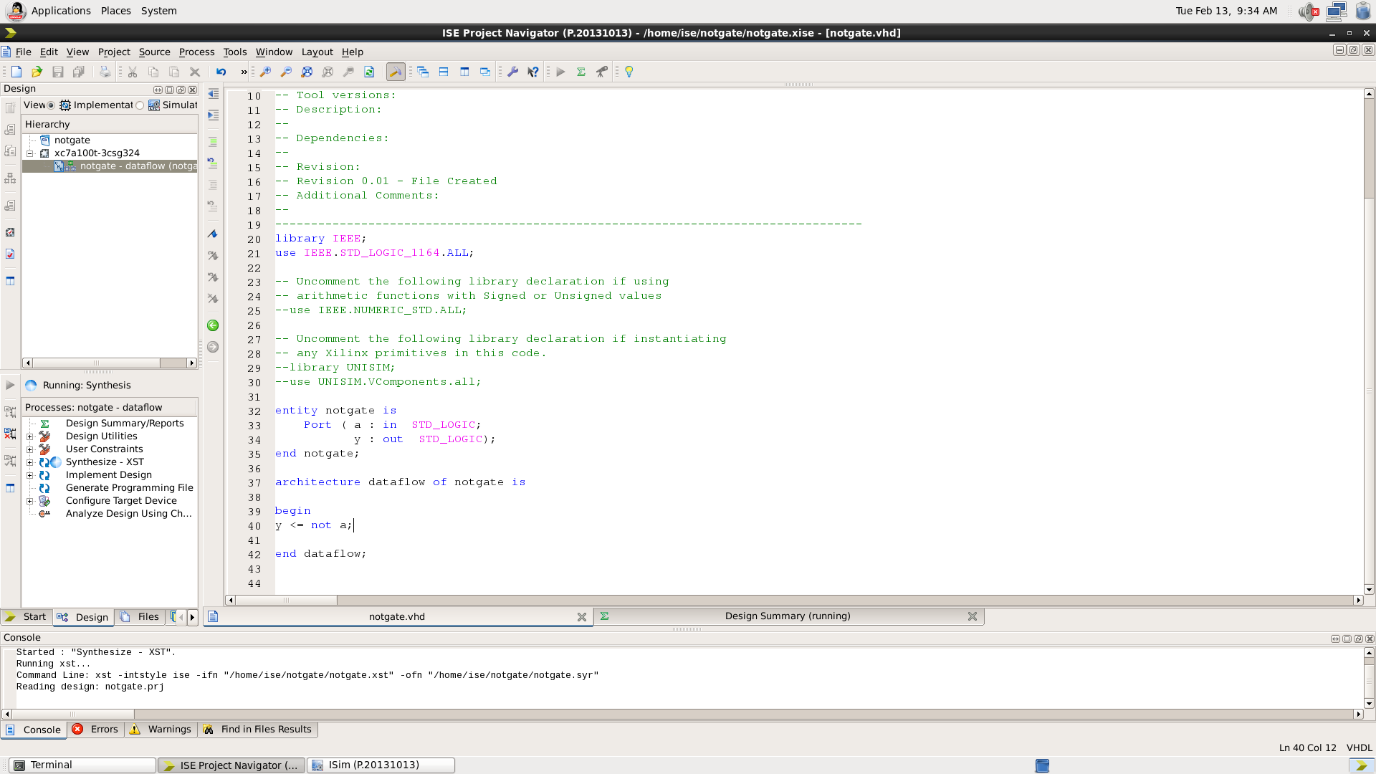
* AND
* OR
* NOT
* NOR (Universal gate\*)
* NAND (Universal gate\*)
* XOR
* XNOR

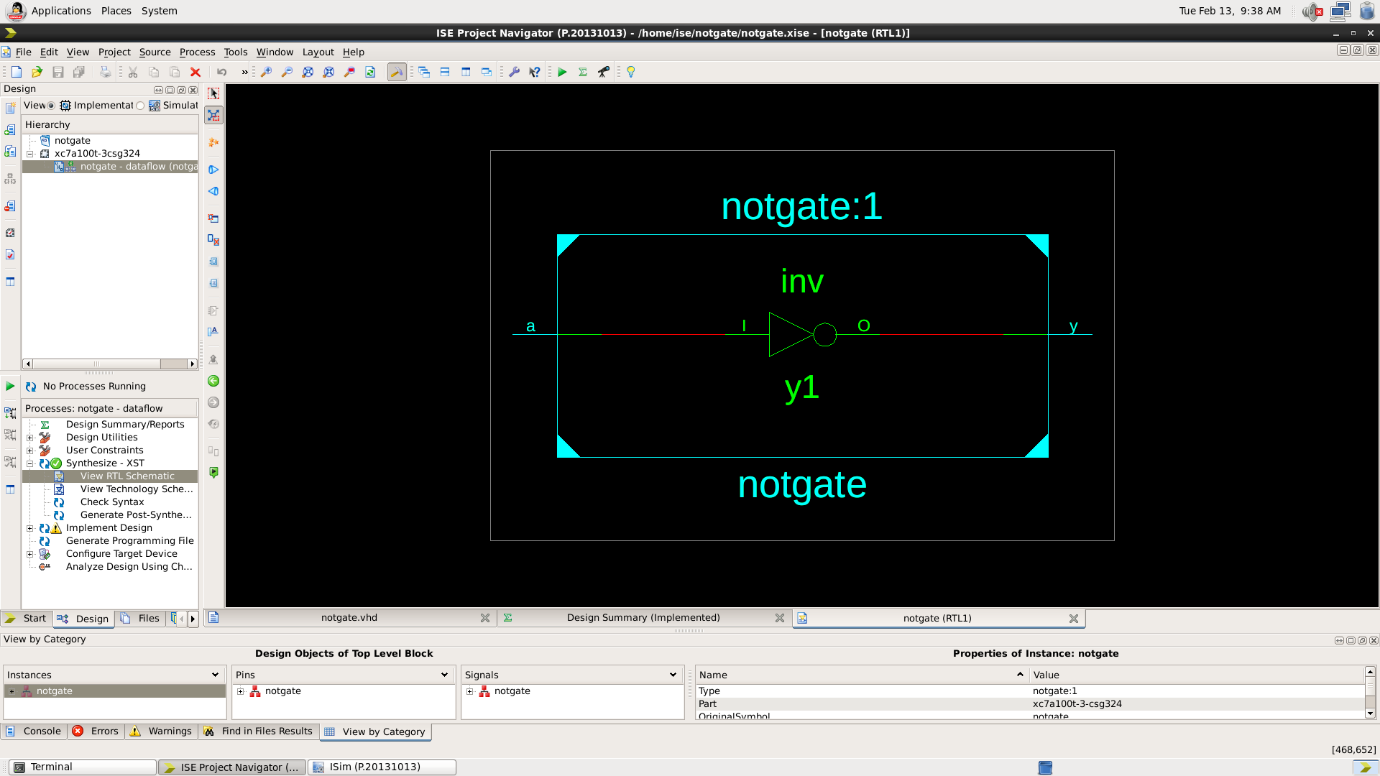
(\* A universal gate is a logic gate which can implement any Boolean function without the need to use any other type of logic gate.)

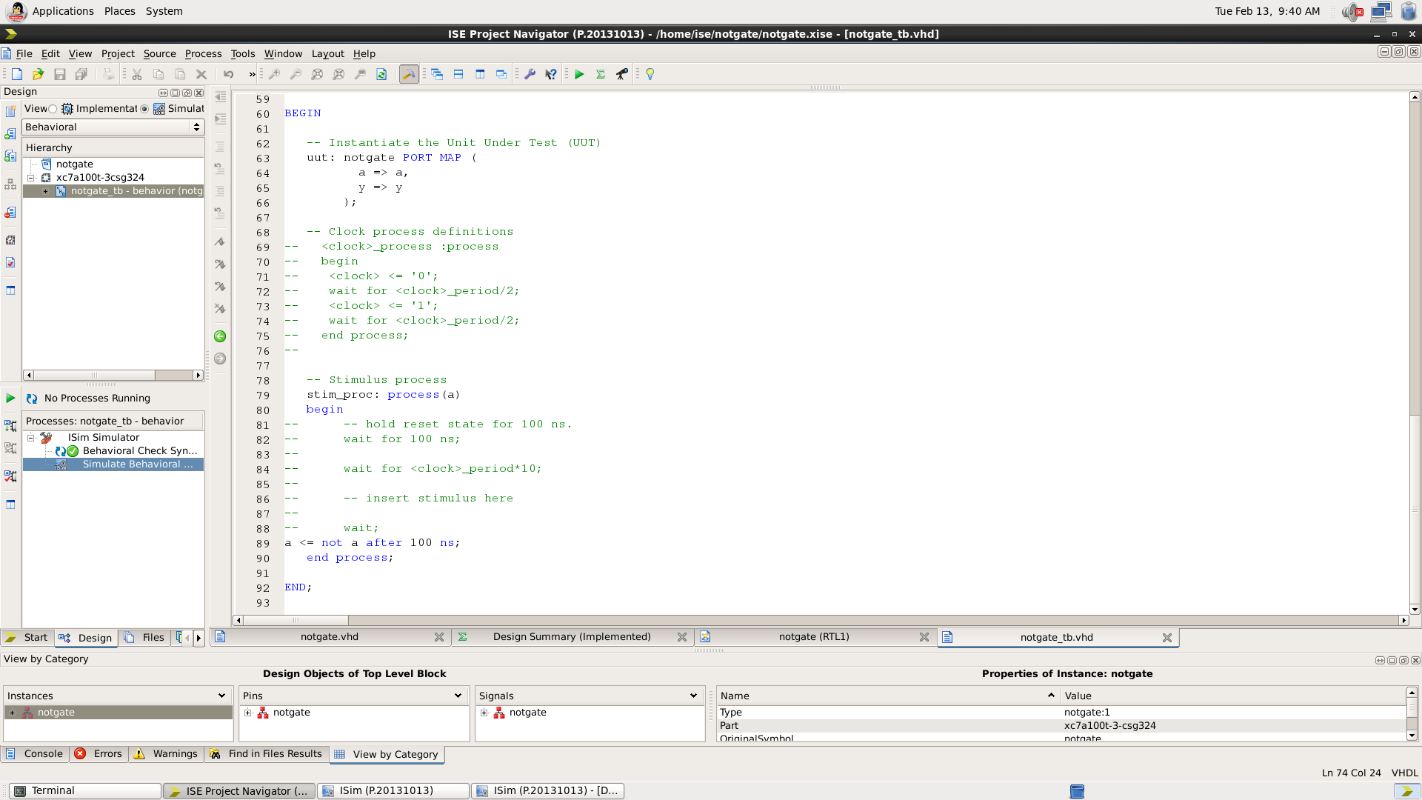
OBSERVATION: The observed outputs of all the basic gates are as follows:

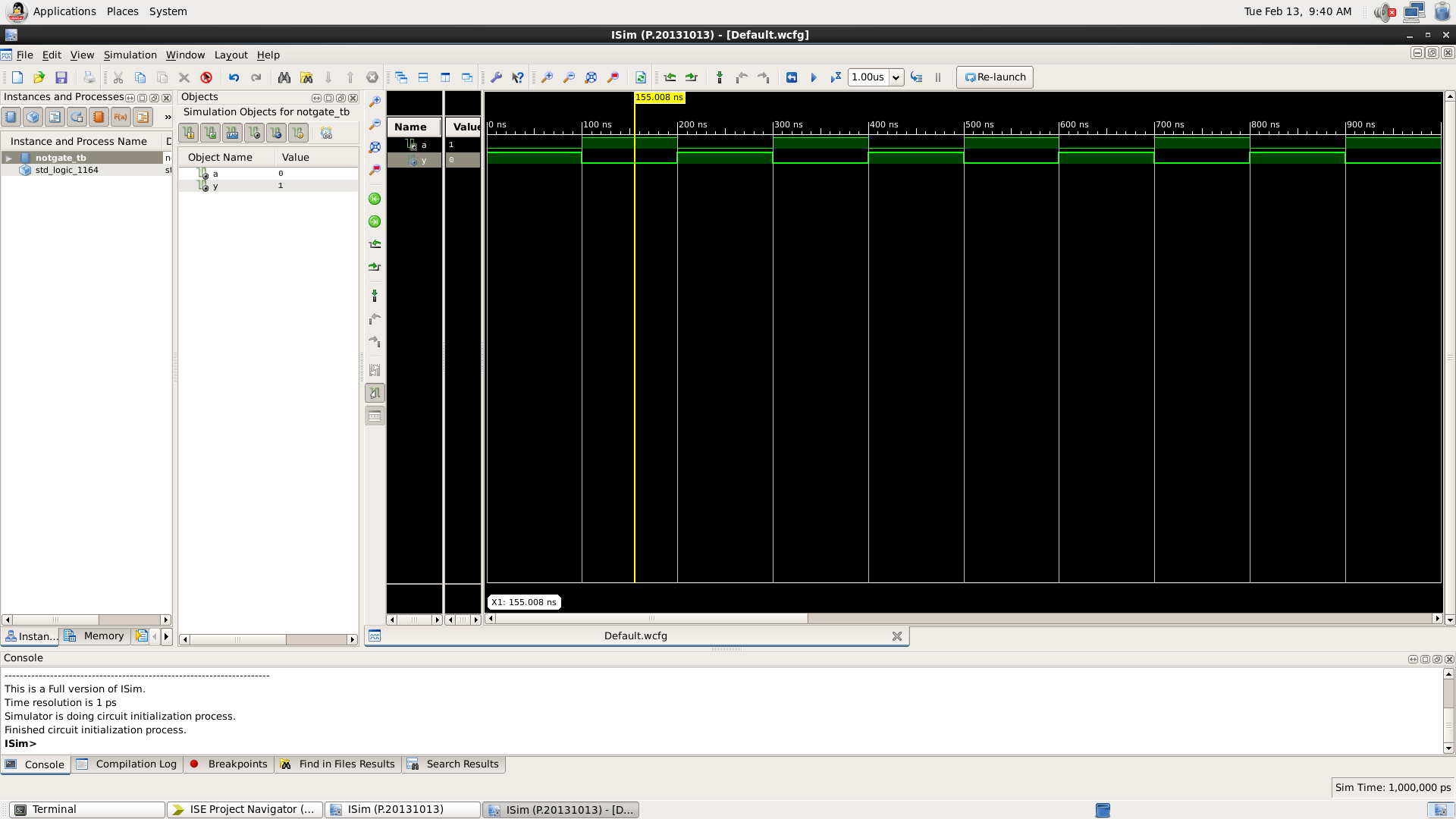
* NOT GATE:

VHDL Code:



RTL Diagram : 

Test Bench Code:

Waveform:

Here the yellow line represents the input - 1 and output -0, which signifies the working of NOT gate. In which we can see the change of input signals after every 100 nano seconds.

|  |  |
| --- | --- |
| INPUT | OUTPUT |
| 1 | 0 |
| 0 | 1 |

Table 1

* AND GATE :

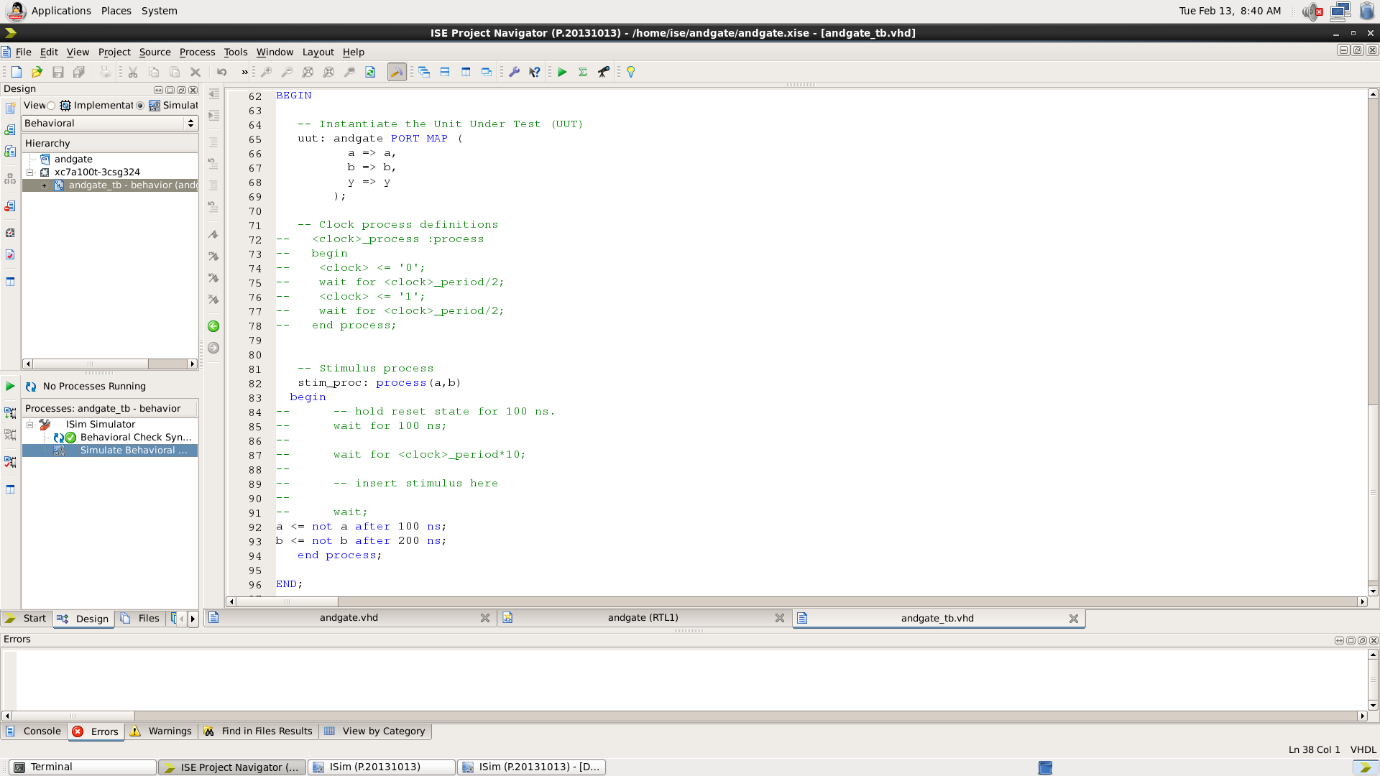
VHDL Code:A screenshot of a computer

Description automatically generated

RTL Diagram:

A computer screen shot of a black screen

Description automatically generated

Test Bench Code:

Waveform:

A screenshot of a computer

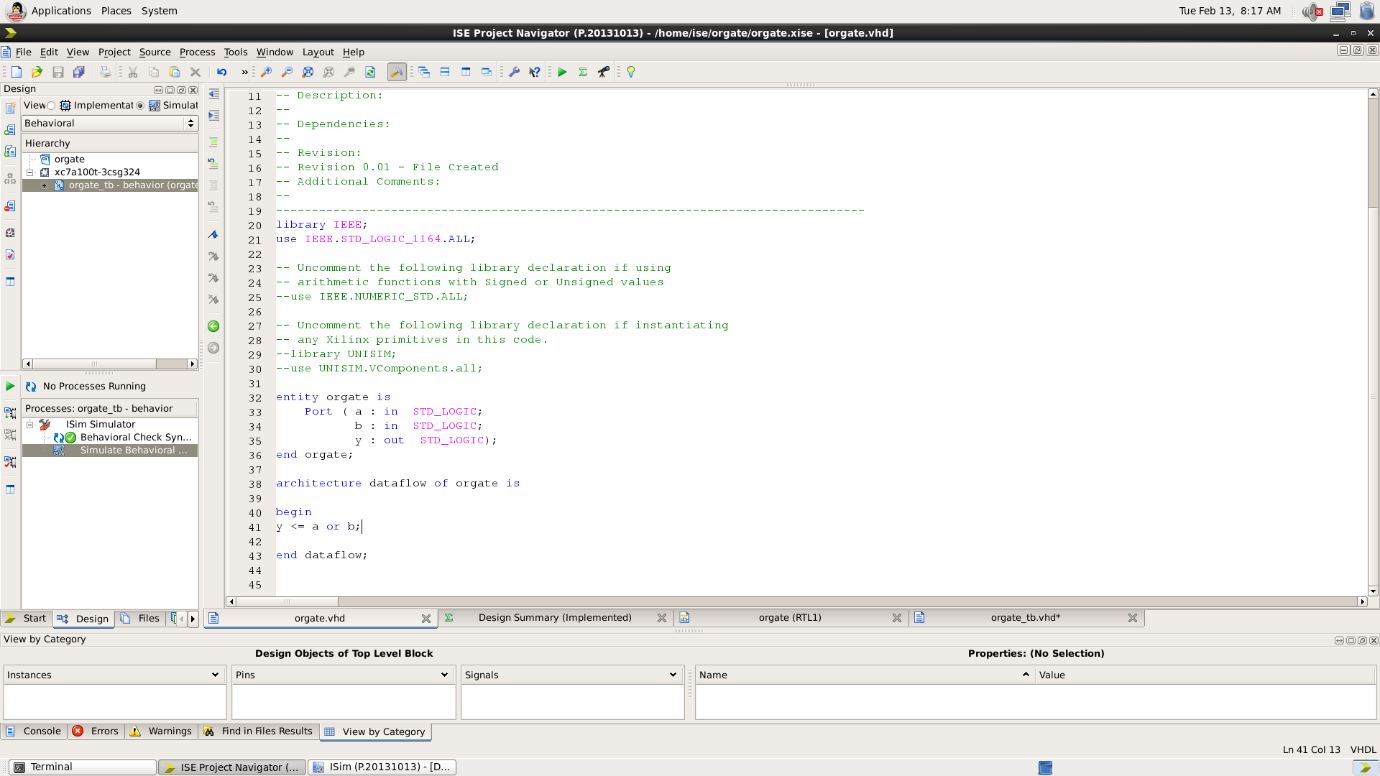
Description automatically generated

Here the yellow line represents the input (a=0 and b=1) and output -0, which signifies the working of AND gate. In which we can see the change of input signals of and b after every 100 and 200 nano seconds respectively.

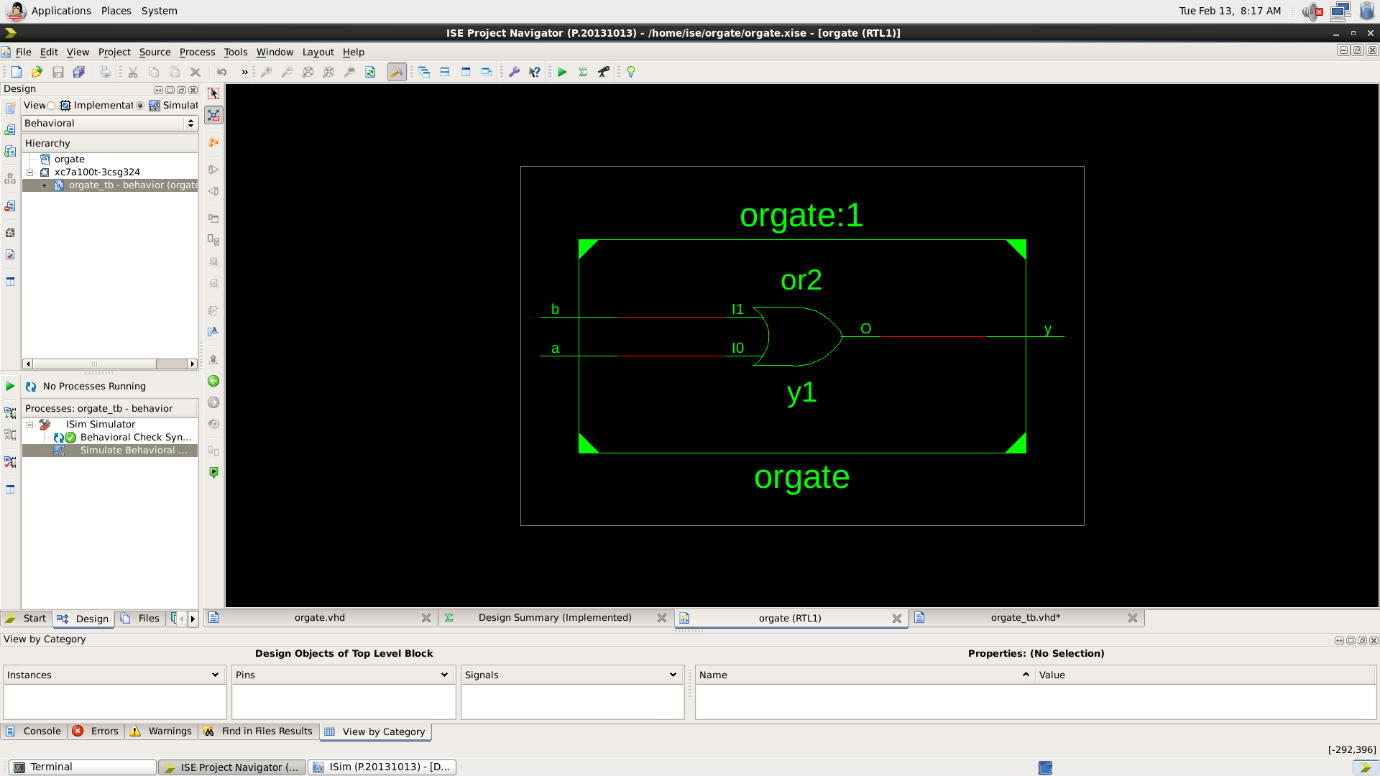
|  |  |  |
| --- | --- | --- |
| A | B | Y = A.B |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

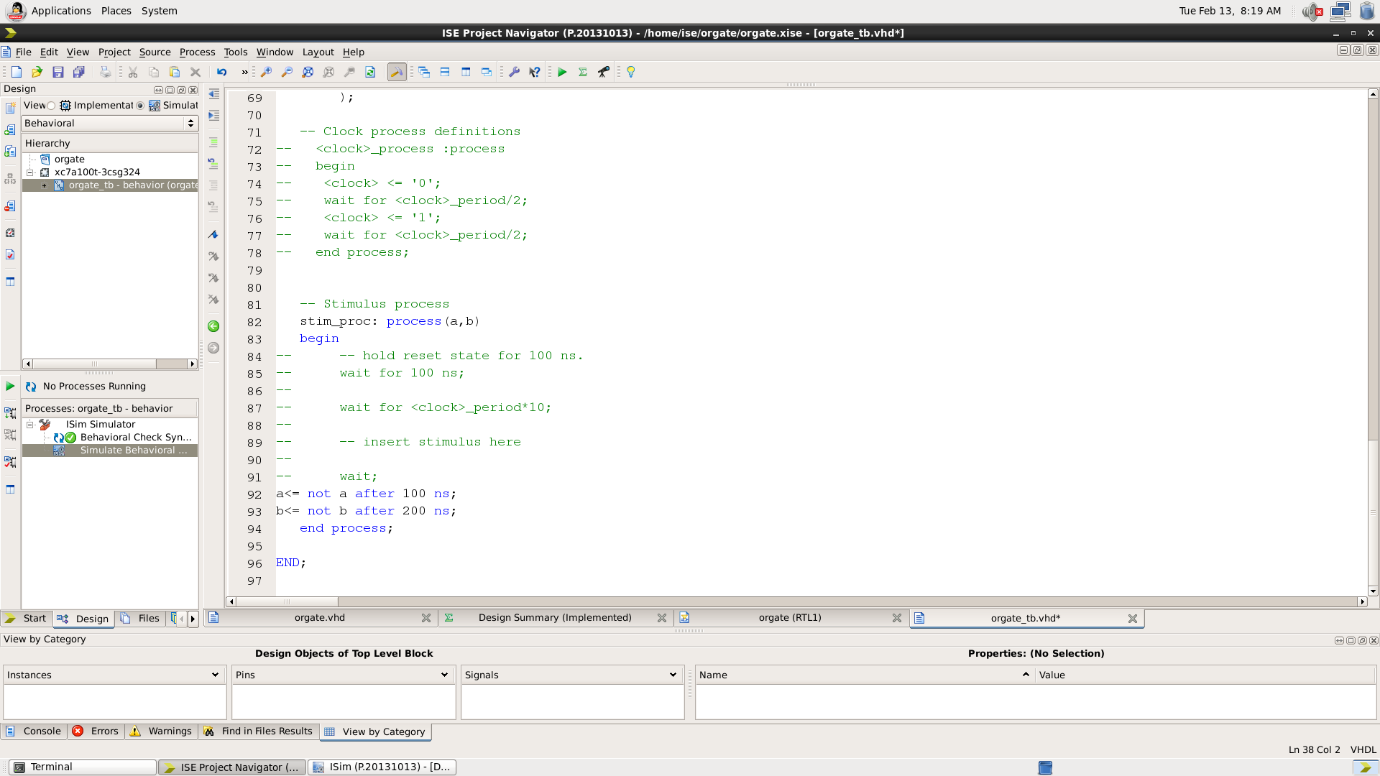
Table 2

* OR GATE:

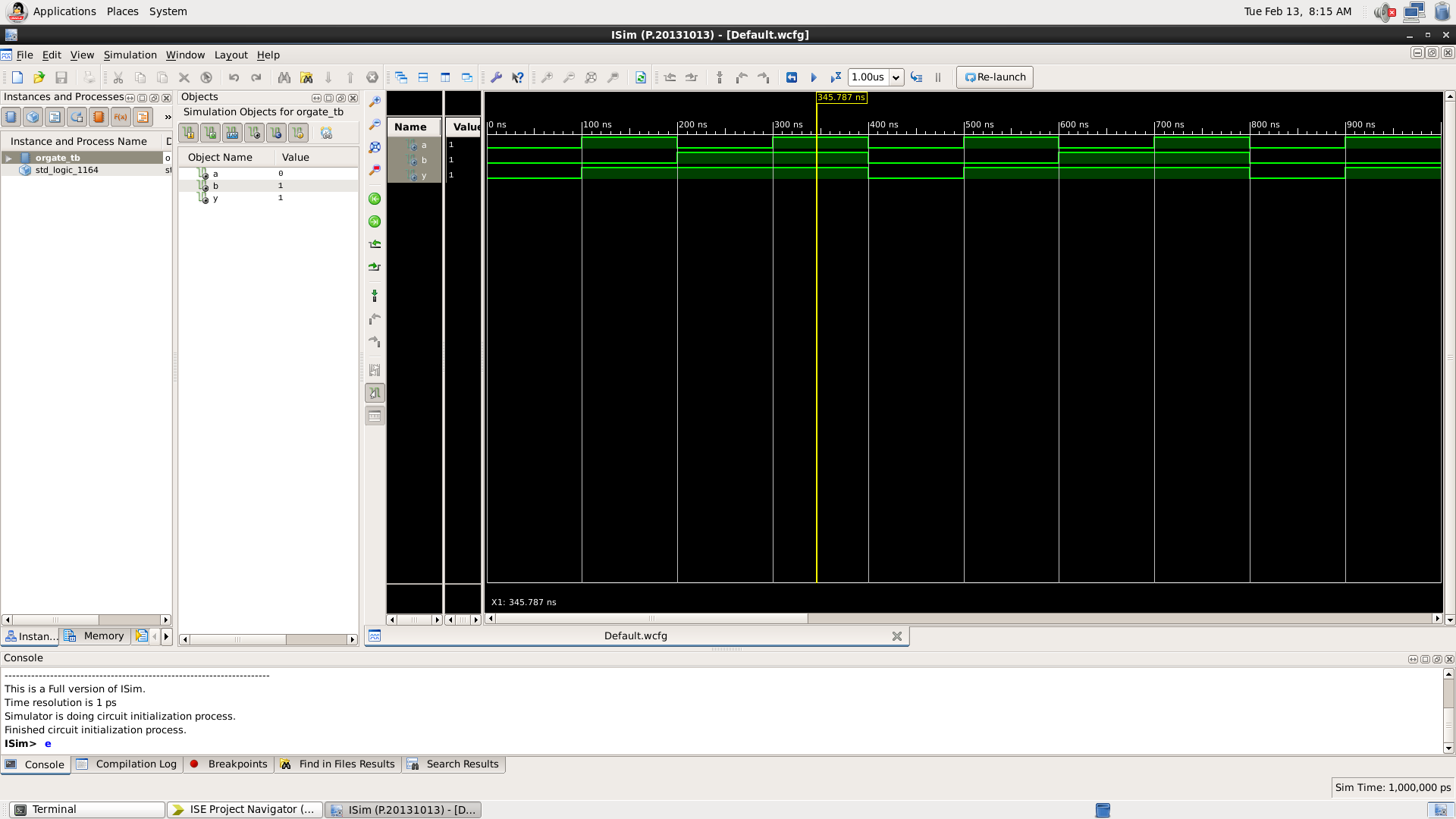
VHDL Code:

RTL Diagram:



Test Bench Code:

Waveform:

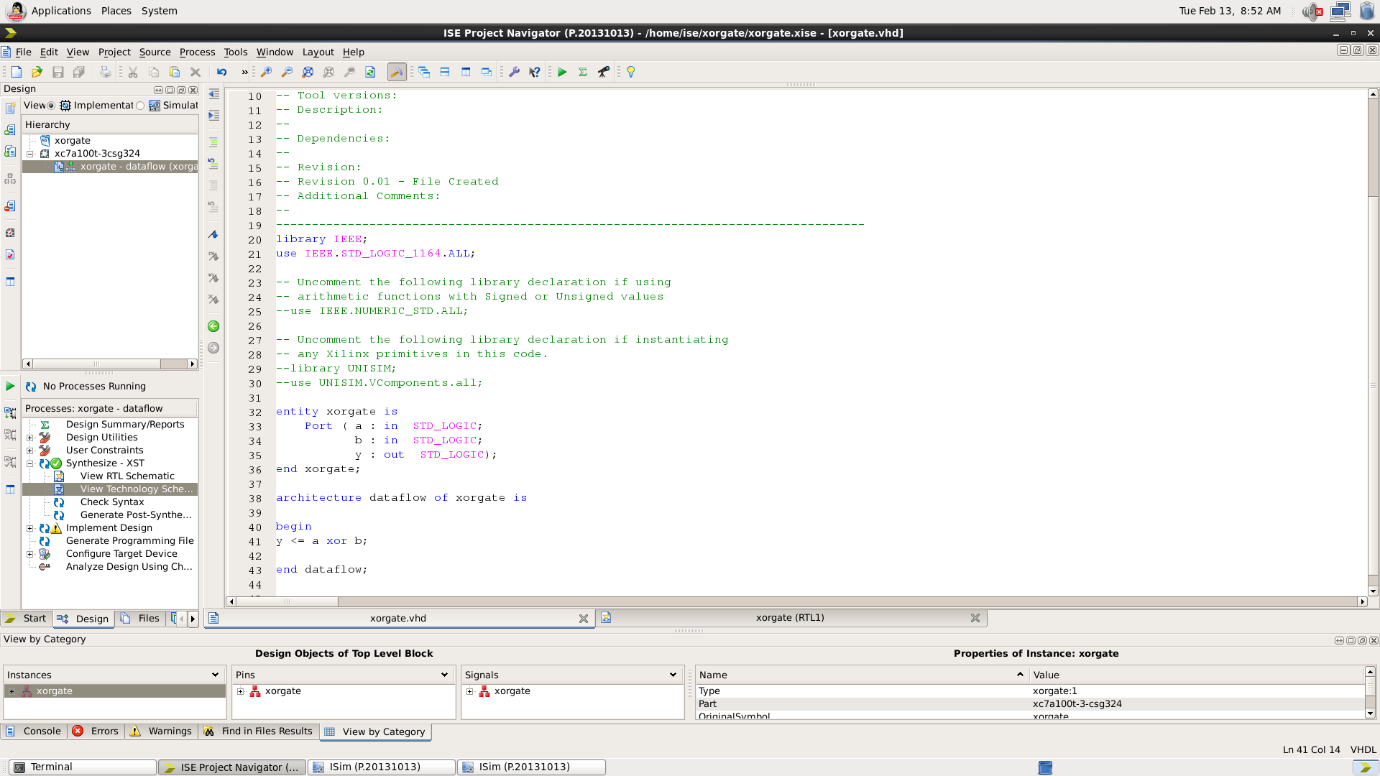


Here the yellow line represents the input (a=1 and b=1) and output -1, which signifies the working of OR gate. In which we can see the change of input signals of and b after every 100 and 200 nano seconds respectively.

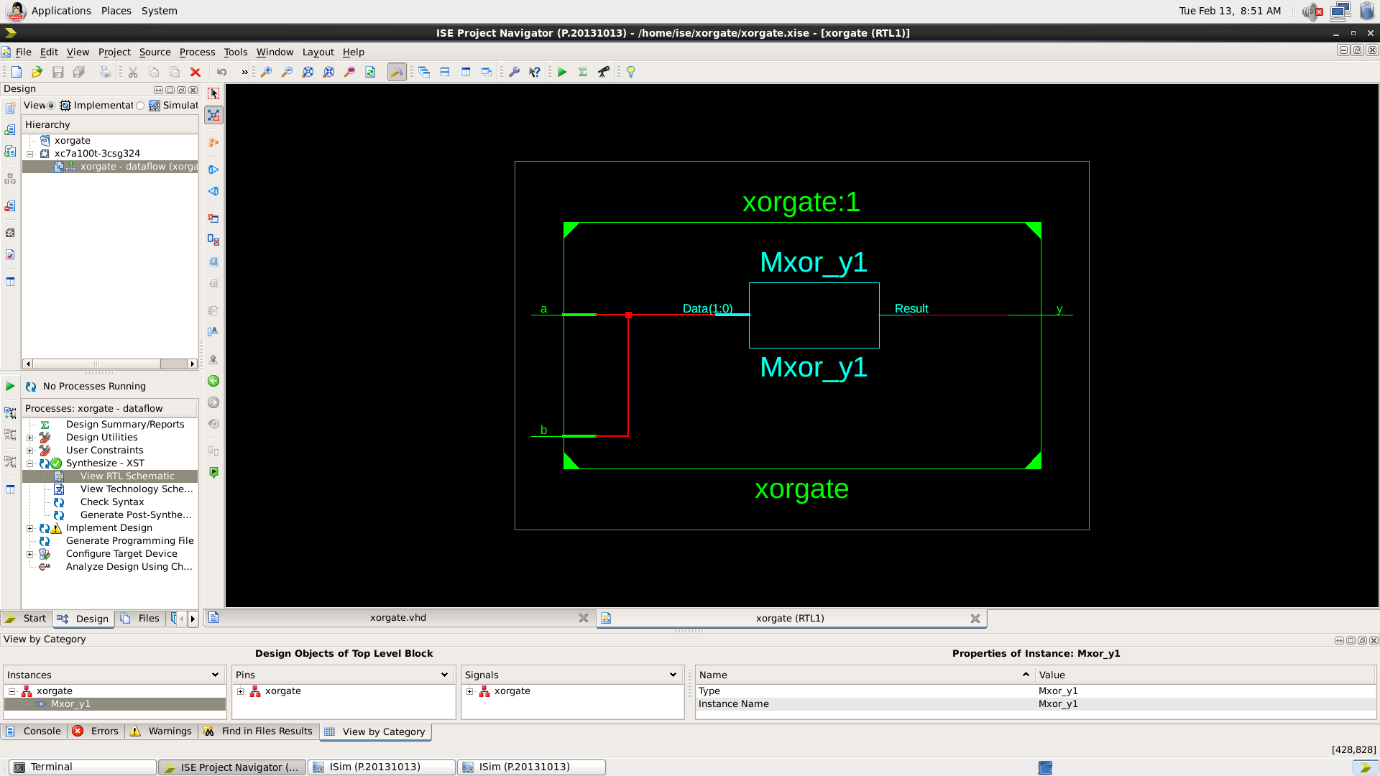
|  |  |  |
| --- | --- | --- |
| A | B | Y = A+B |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

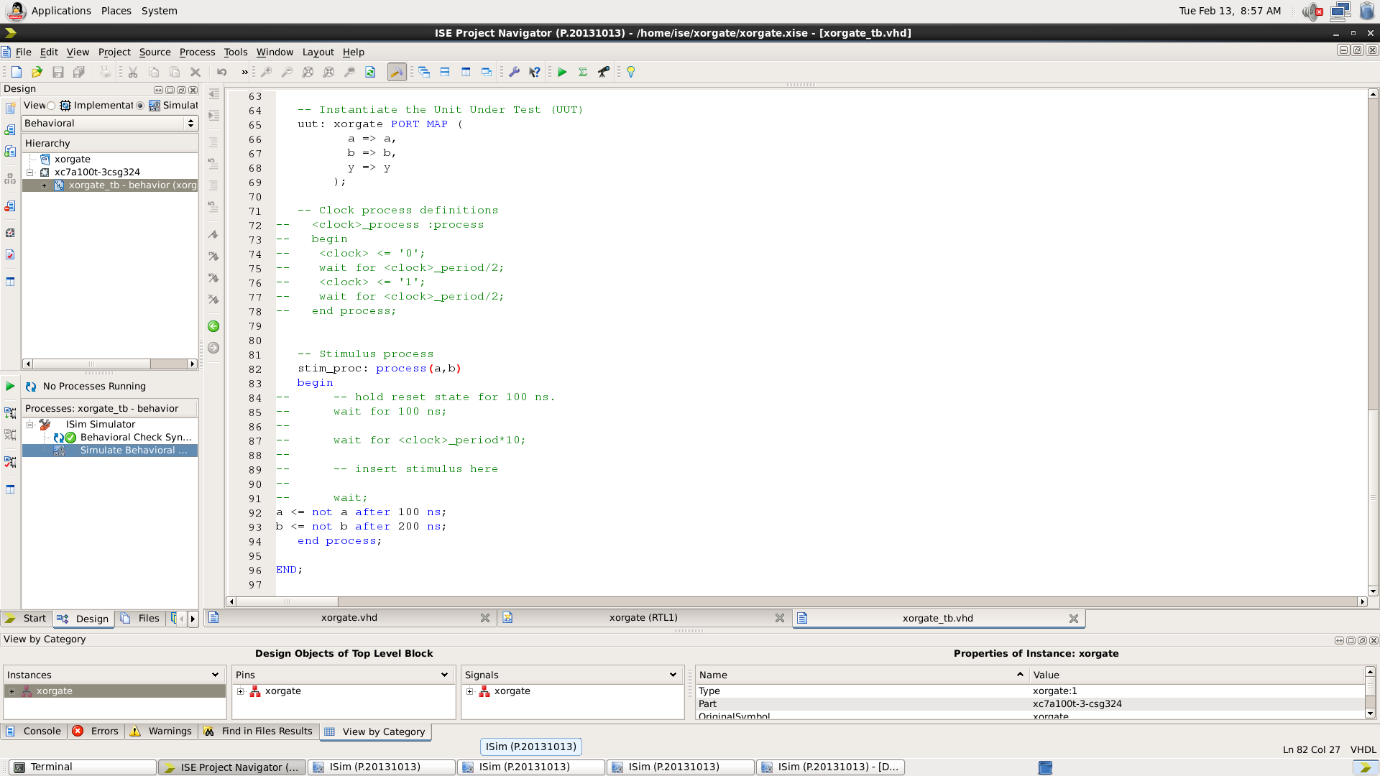
Table 3

* X-OR GATE:

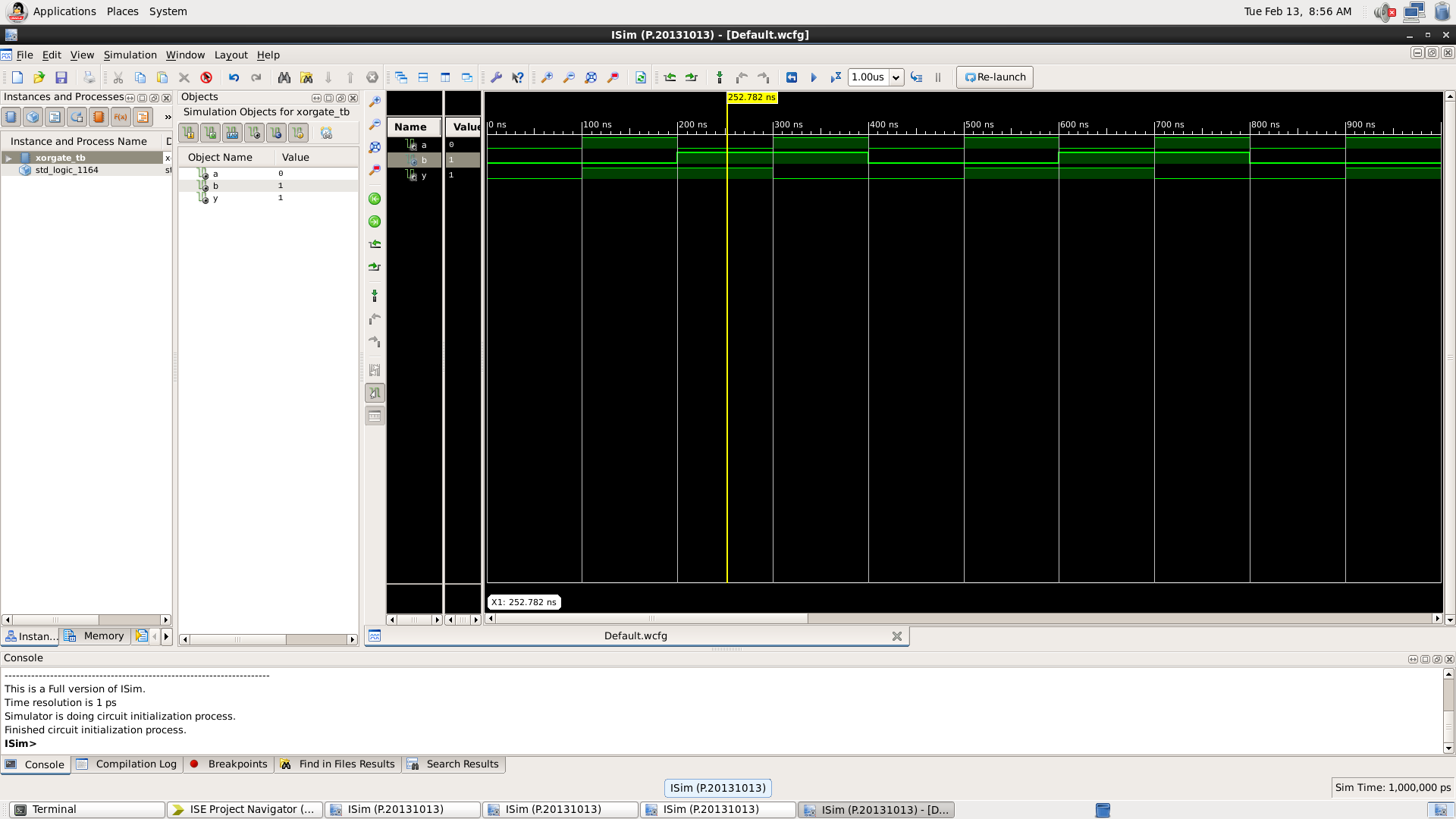
VHDL Code:

RTL Diagram:



Test Bench Code:

Waveform:

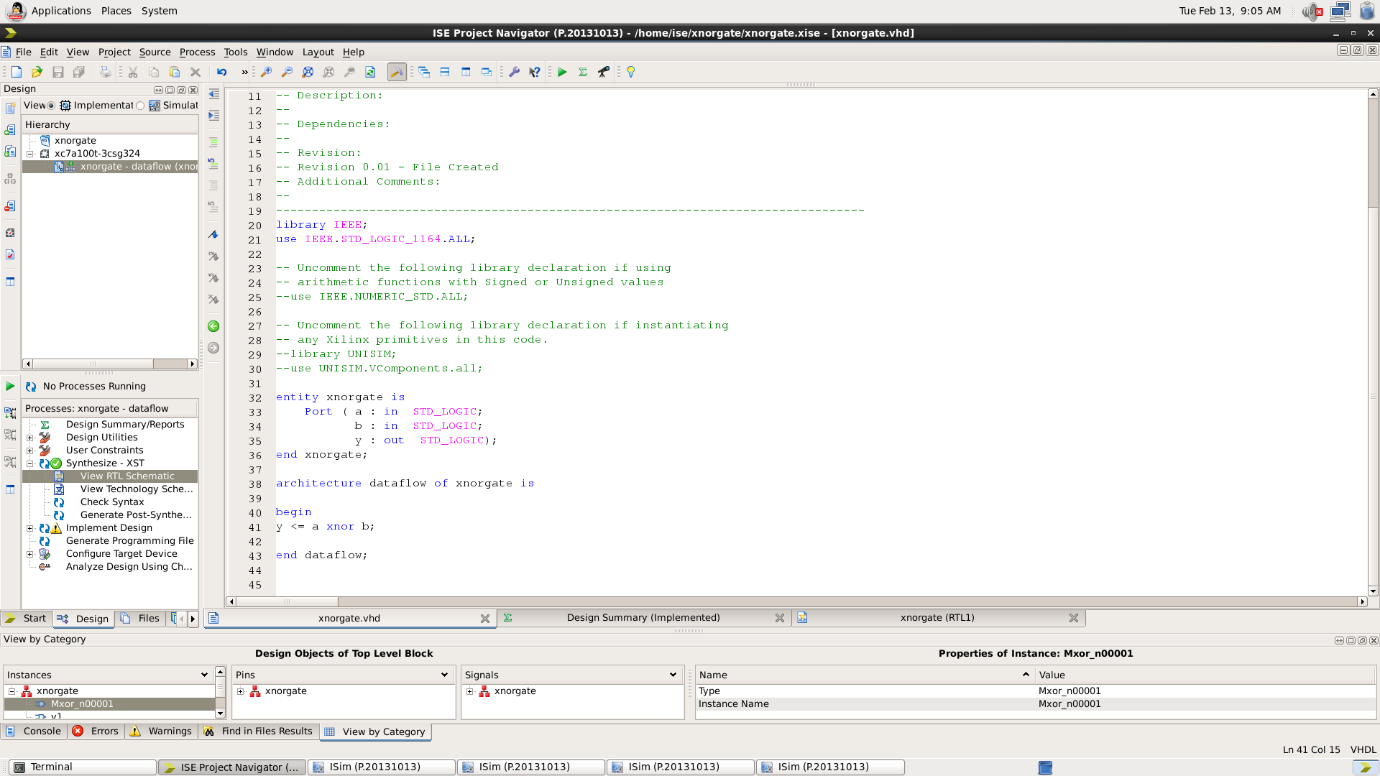


Here the yellow line represents the input (a=0 and b=1) and output -1, which signifies the working of XOR gate. In which we can see the change of input signals of and b after every 100 and 200 nano seconds respectively.

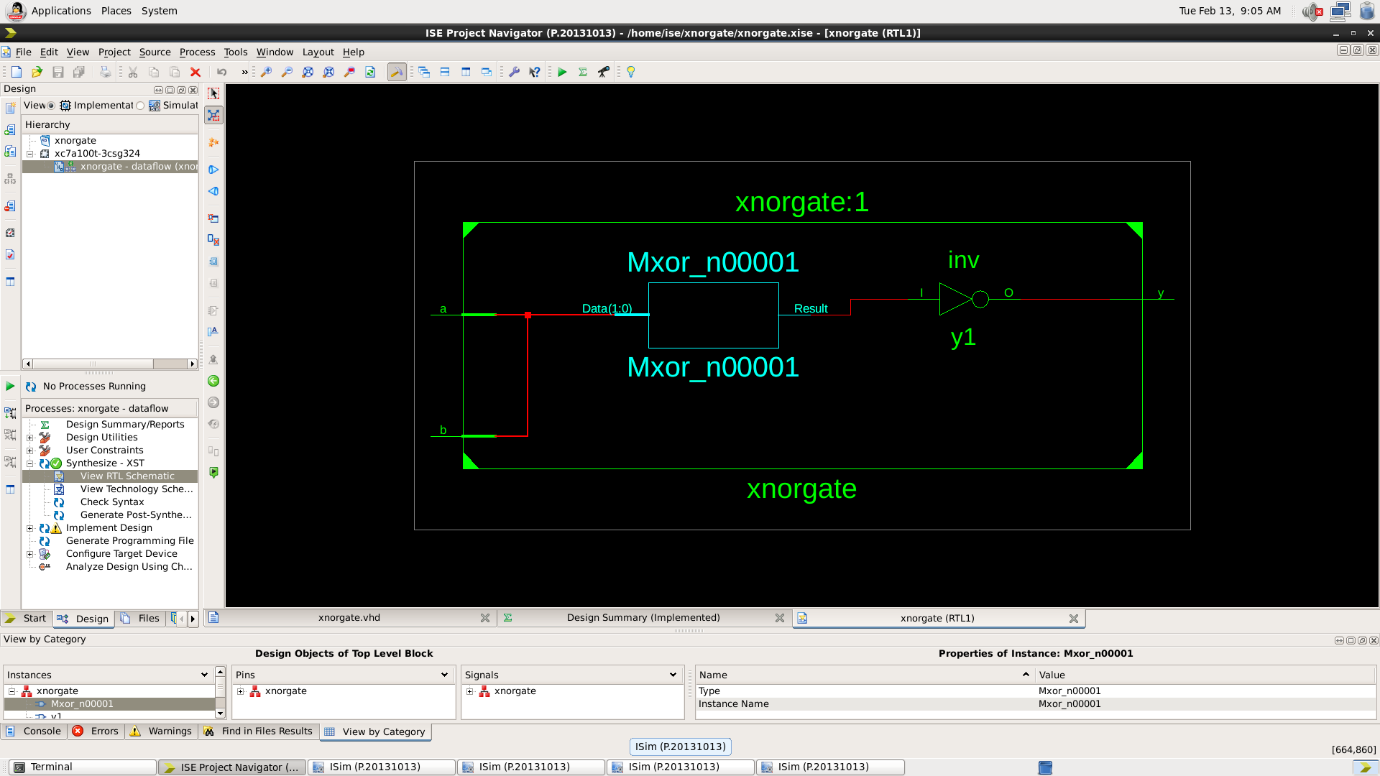
|  |  |  |
| --- | --- | --- |
| A | B | Y = A+B (Exclusive or) |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

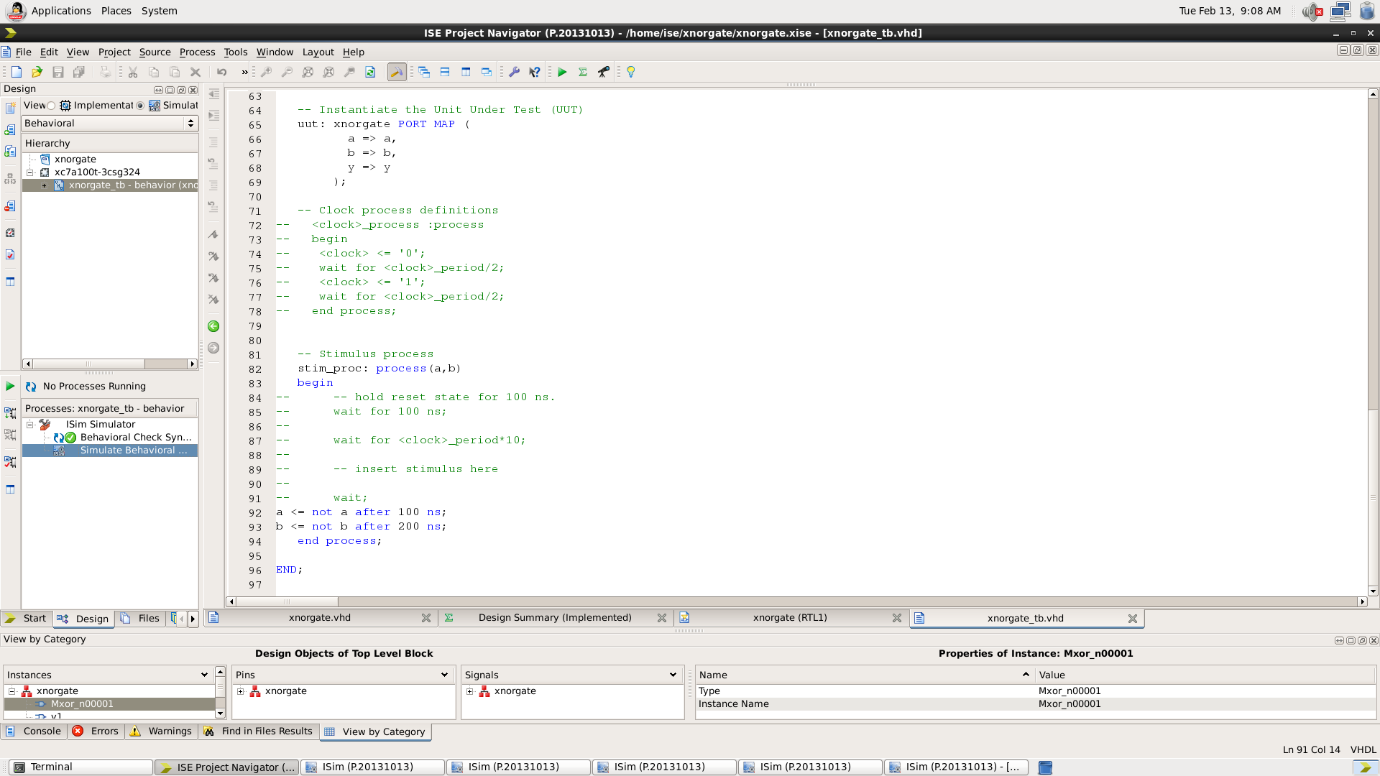
Table 4

* X-NOR GATE:

VHDL Code:

RTL Diagram:



Test Bench Code:

Waveform:

A screenshot of a computer

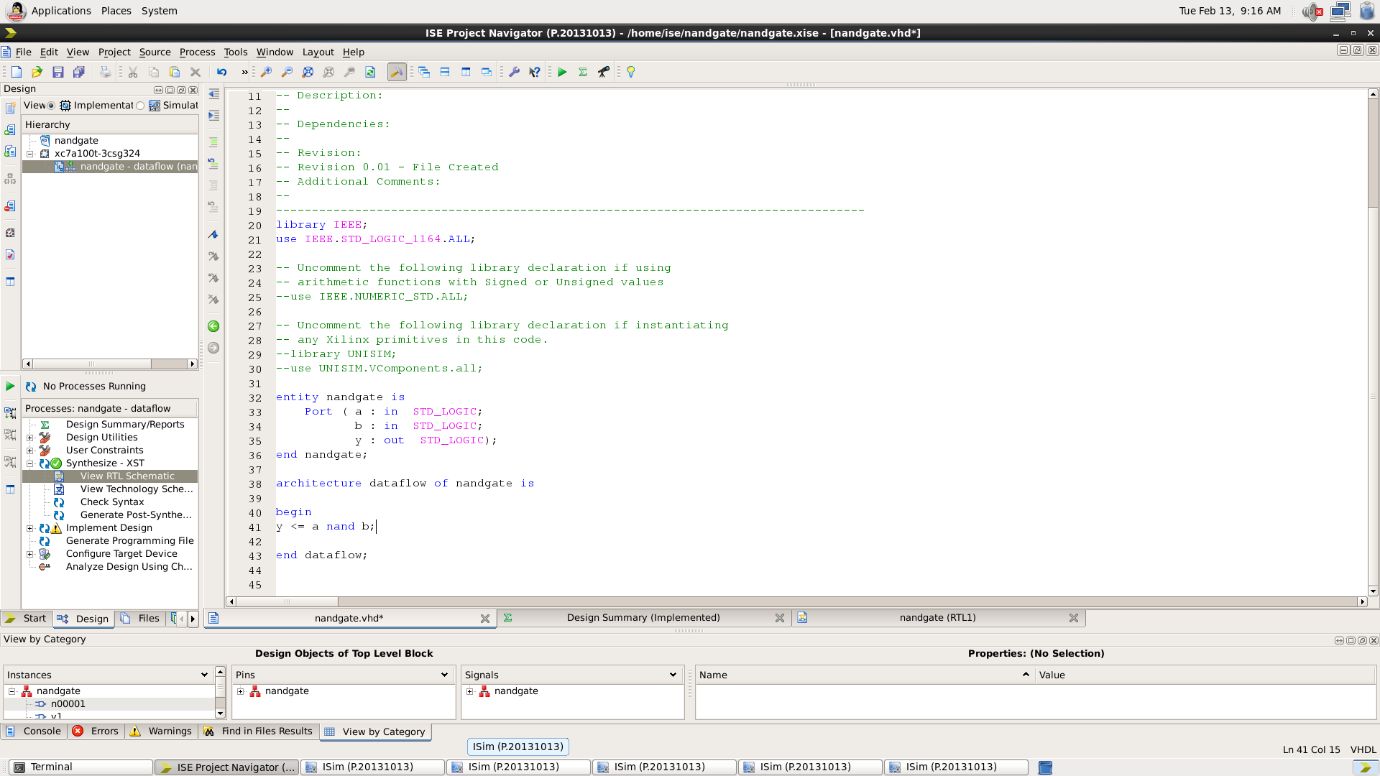
Description automatically generated

Here the yellow line represents the input (a=1 and b=1) and output -1, which signifies the working of XNOR gate. In which we can see the change of input signals of and b after every 100 and 200 nano seconds respectively.

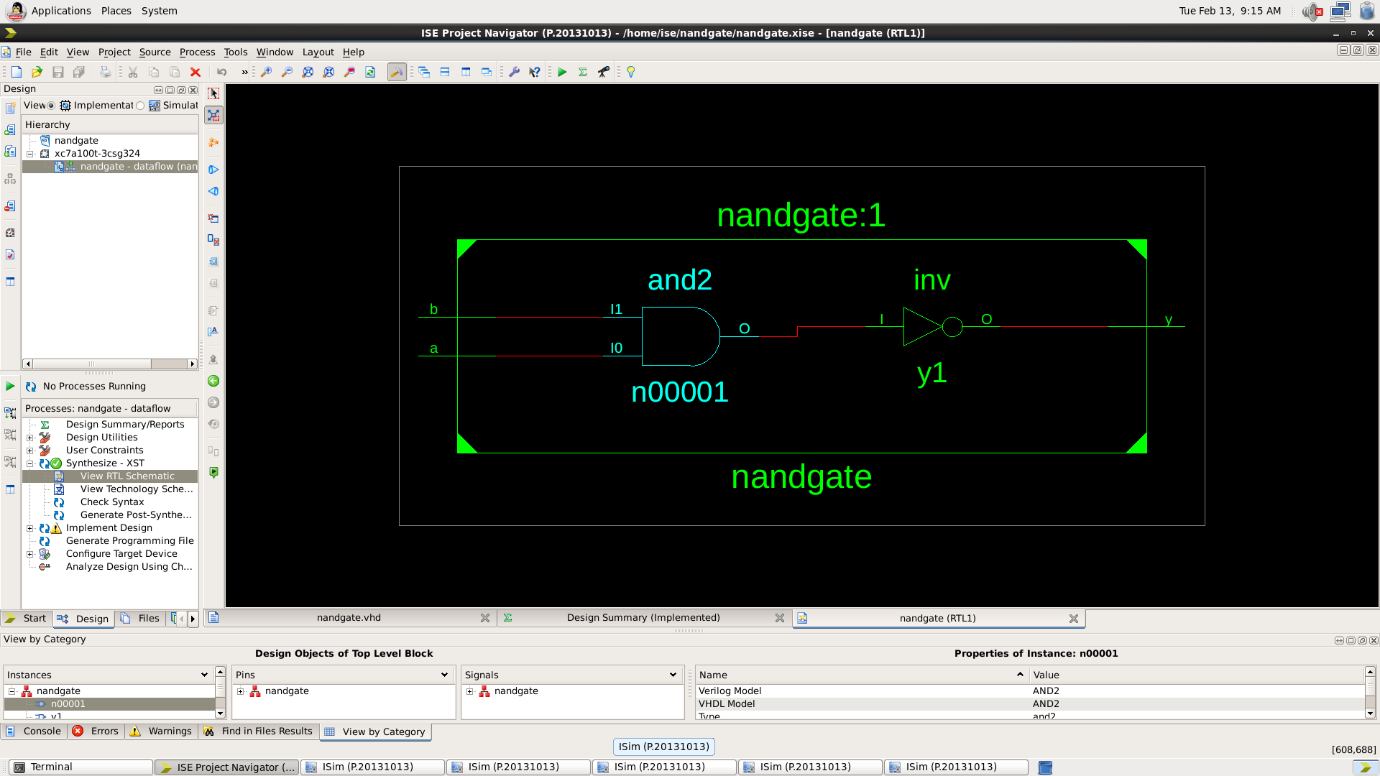
|  |  |  |
| --- | --- | --- |
| A | B | Y = Complement of XOR gate |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

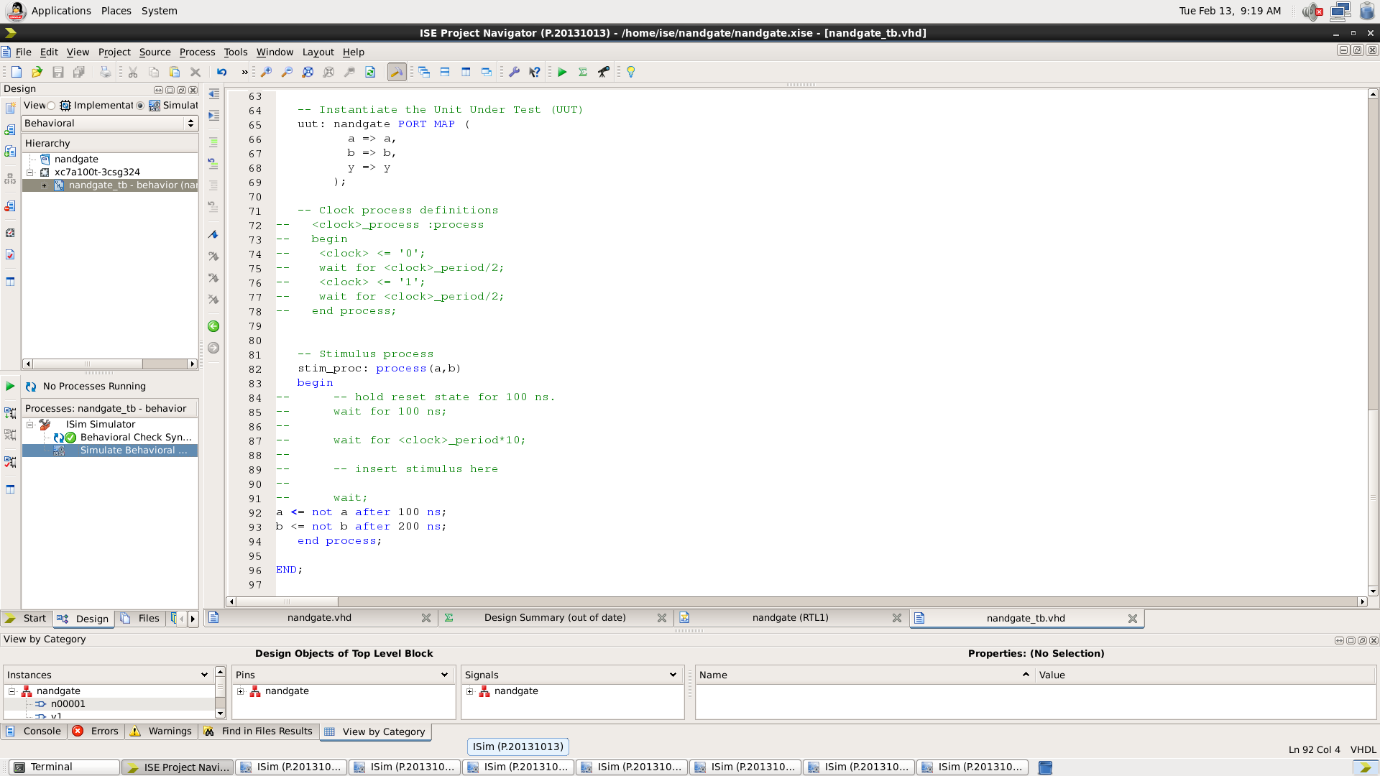
Table 5

* NAND GATE:

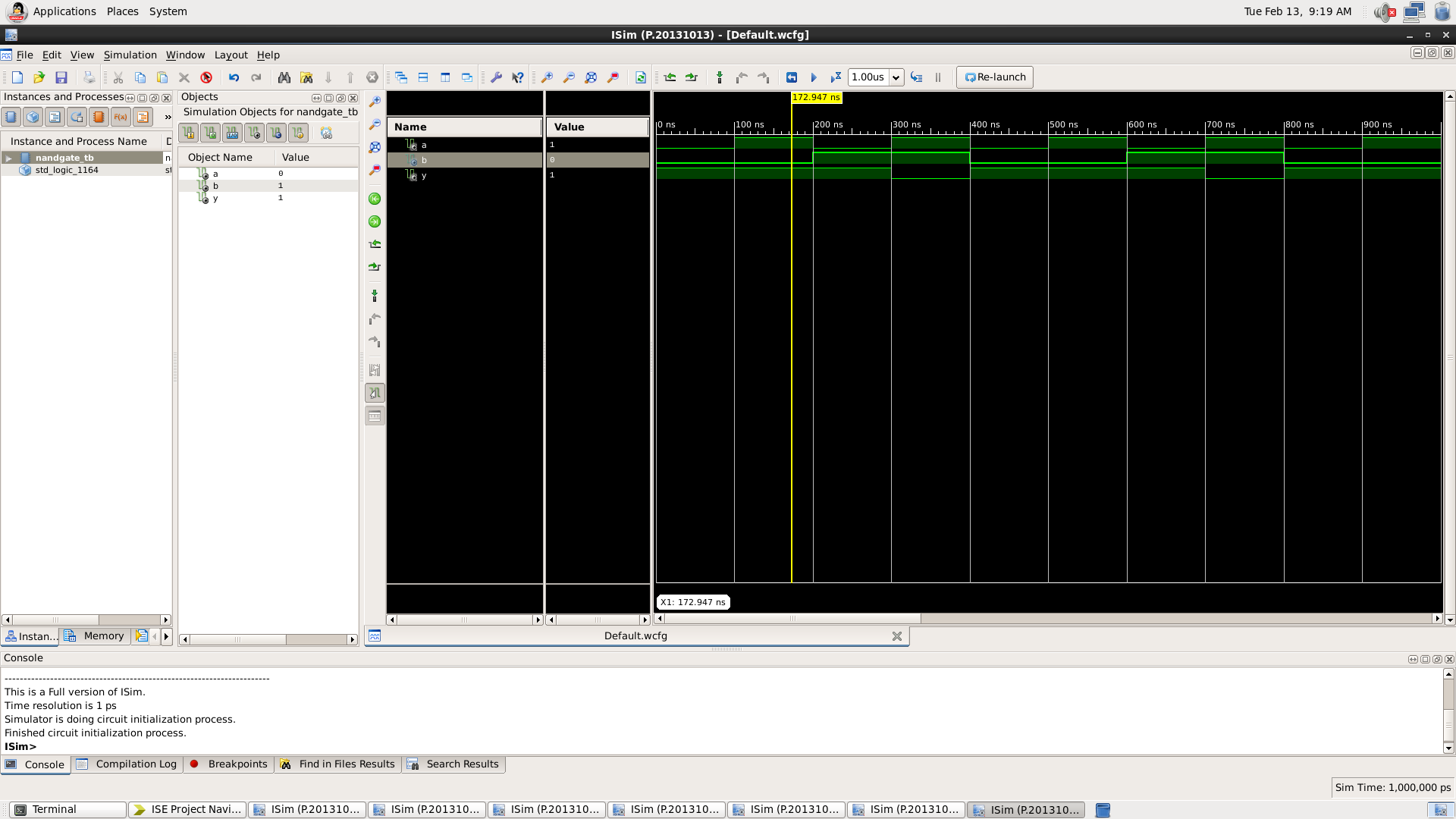
VHDL Code:

RTL Diagram:



Test Bench Code:

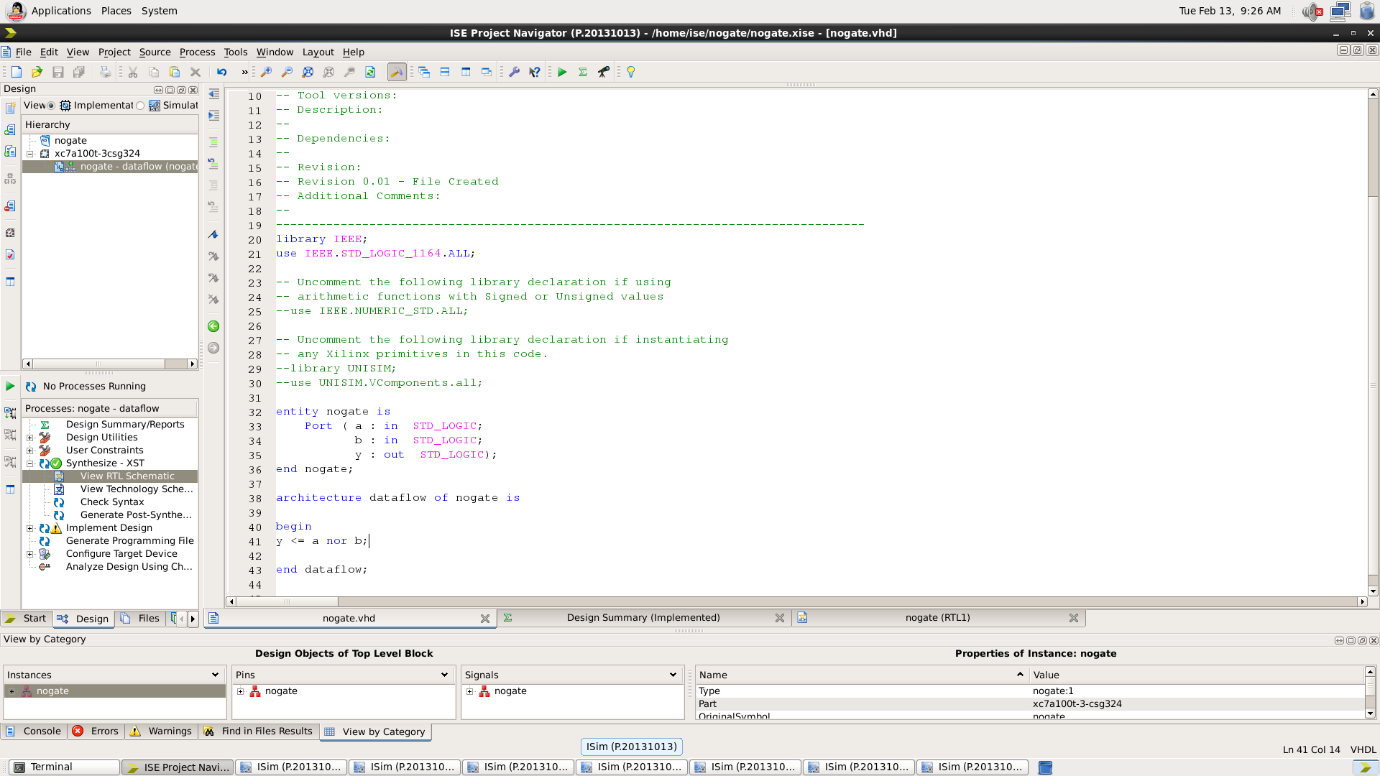
Waveform:



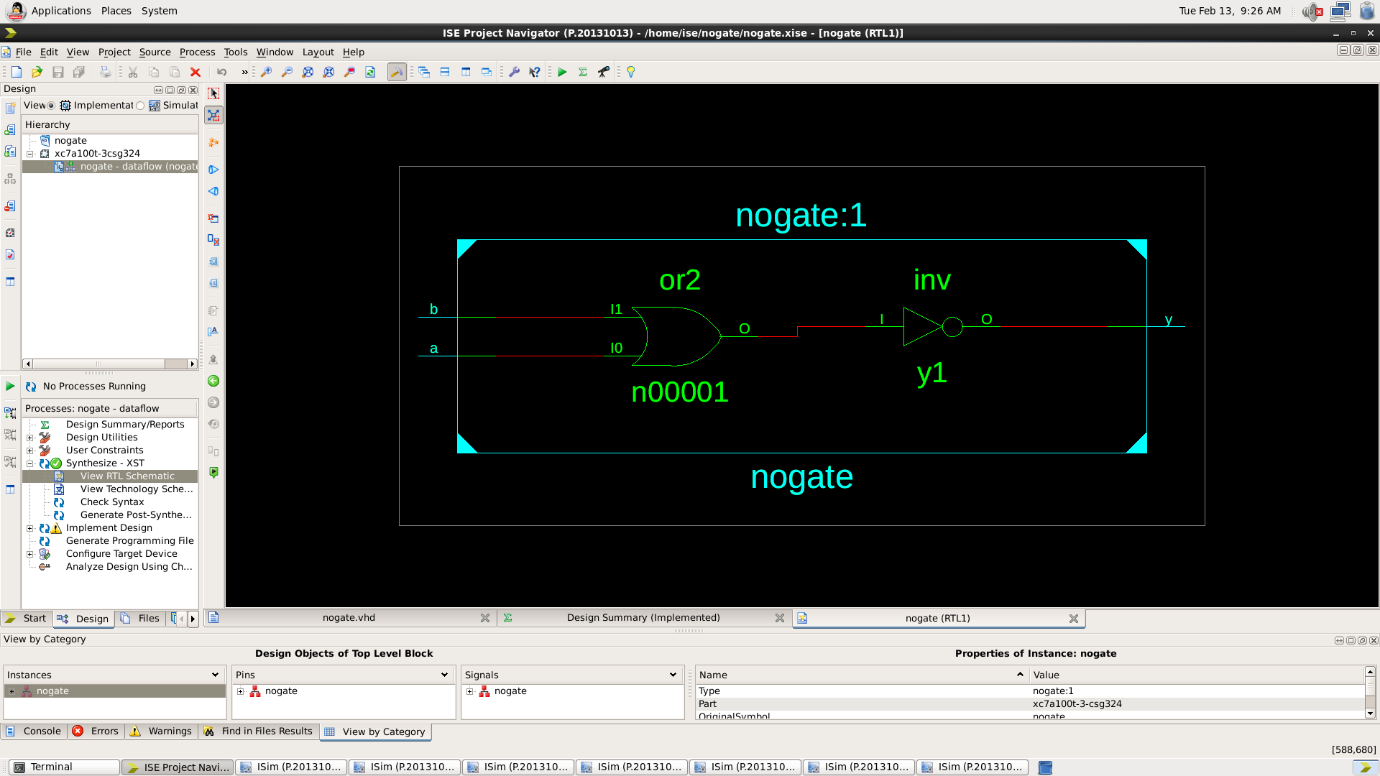
Here the yellow line represents the input (a=1 and b=0) and output -1, which signifies the working of NAND gate. In which we can see the change of input signals of and b after every 100 and 200 nano seconds respectively.

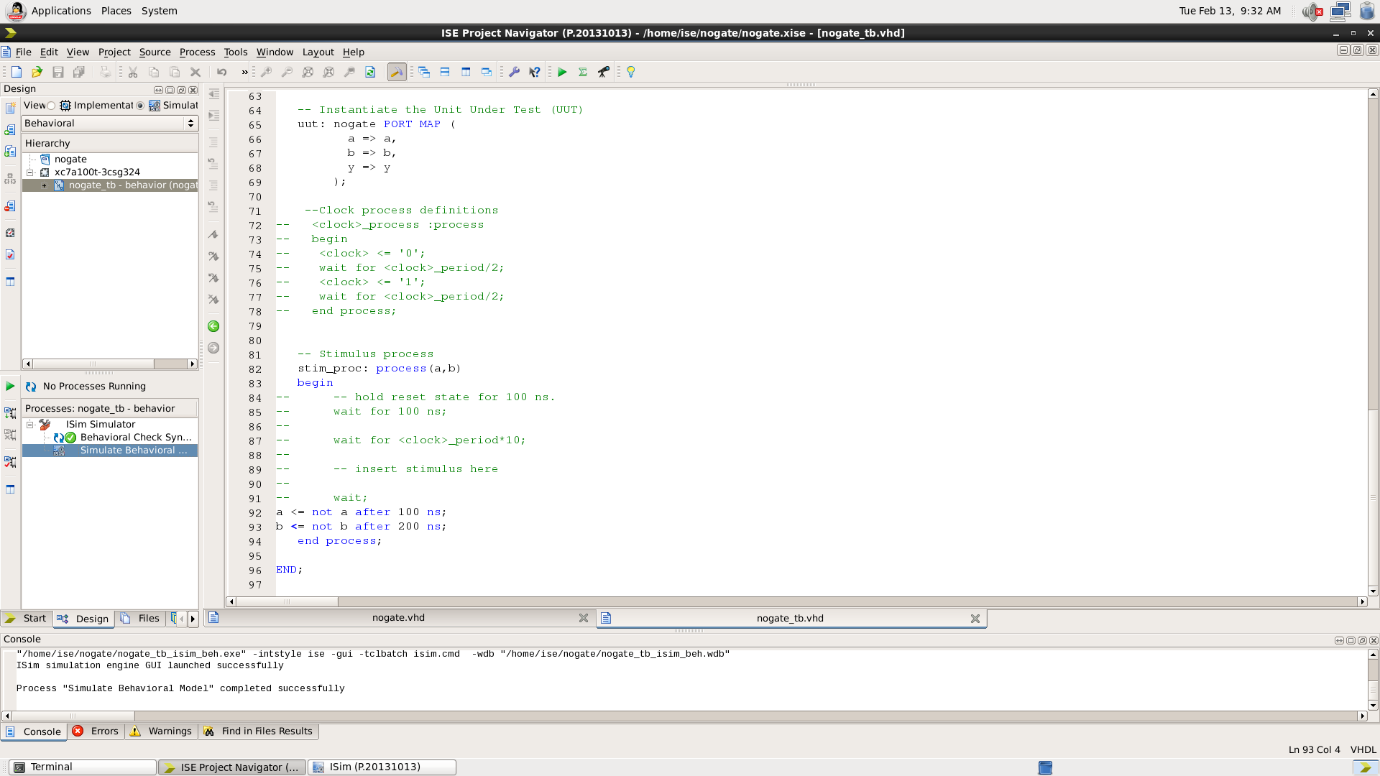
|  |  |  |
| --- | --- | --- |
| A | B | Y = Complement of AND gate |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

* NOR GATE:

VHDL Code:

RTL Diagram:



Test Bench Code:

Waveform:

A screenshot of a computer

Description automatically generated

Here the yellow line represents the input (a=1 and b=0) and output -0, which signifies the working of NOR gate. In which we can see the change of input signals of and b after every 100 and 200 nano seconds respectively.

|  |  |  |
| --- | --- | --- |
| A | B | Y = Complement of OR gate |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

RESULT: We have concluded all the truth tables of all the basic logic gates using tinker cad simulation.

APPLICATION IN DAILY LIFE:

VHDL (VHSIC Hardware Description Language) is primarily used for the description and simulation of digital circuits and systems. While it is more commonly associated with the field of digital design and hardware engineering, its impact can be indirectly observed in various aspects of daily life through the products and technologies that rely on digital systems. Here are some applications and areas where VHDL plays a role:

* Consumer Electronics: VHDL is used in the design and verification of digital components in consumer electronics such as smartphones, tablets, TVs, and audio devices. The digital signal processing, control logic, and communication interfaces often involve VHDL in their design.
* Automotive Systems: Modern vehicles incorporate numerous digital systems for engine control, safety features, infotainment, and more. VHDL is employed in the design and testing of the digital components within these systems.
* Communication Systems: VHDL is used in the development of communication systems, including network routers, switches, and wireless communication devices. It plays a crucial role in the design of digital signal processing algorithms and communication protocols.