**ASSIGNMENT- I**

**NAME: SOUMYADIP GHOSH**

**STREAM: CSE- A**

**ROLL NUMBER: 1951007**

**SUBJECT: COMPUTER ARCHITECTURE LAB**

**AND GATE**

**TRUTH TABLE:**

|  |  |  |
| --- | --- | --- |
| Input | | Output |
| X | Y | Z |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**DESIGN SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity testand1 is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end testand1;

architecture Dataflow of testand1 is

begin

Z <= X AND Y;

end Dataflow;

**SIMULATION SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity testand1\_tb is

-- Port ( );

end testand1\_tb;

architecture Behavioral of testand1\_tb is

component testand1 is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end component;

signal X1 : STD\_LOGIC := '0';

signal Y1 : STD\_LOGIC := '0';

signal Z1 : STD\_LOGIC;

begin

uut: testand1 port map (X=>X1, Y=>Y1, Z=>Z1);

stim\_proc: process

begin

wait for 100ns;

X1 <= '0';

Y1 <= '0';

wait for 100ns;

X1 <= '0';

Y1 <= '1';

wait for 100ns;

X1 <= '1';

Y1 <= '0';

wait for 100ns;

X1 <= '1';

Y1 <= '1';

wait;

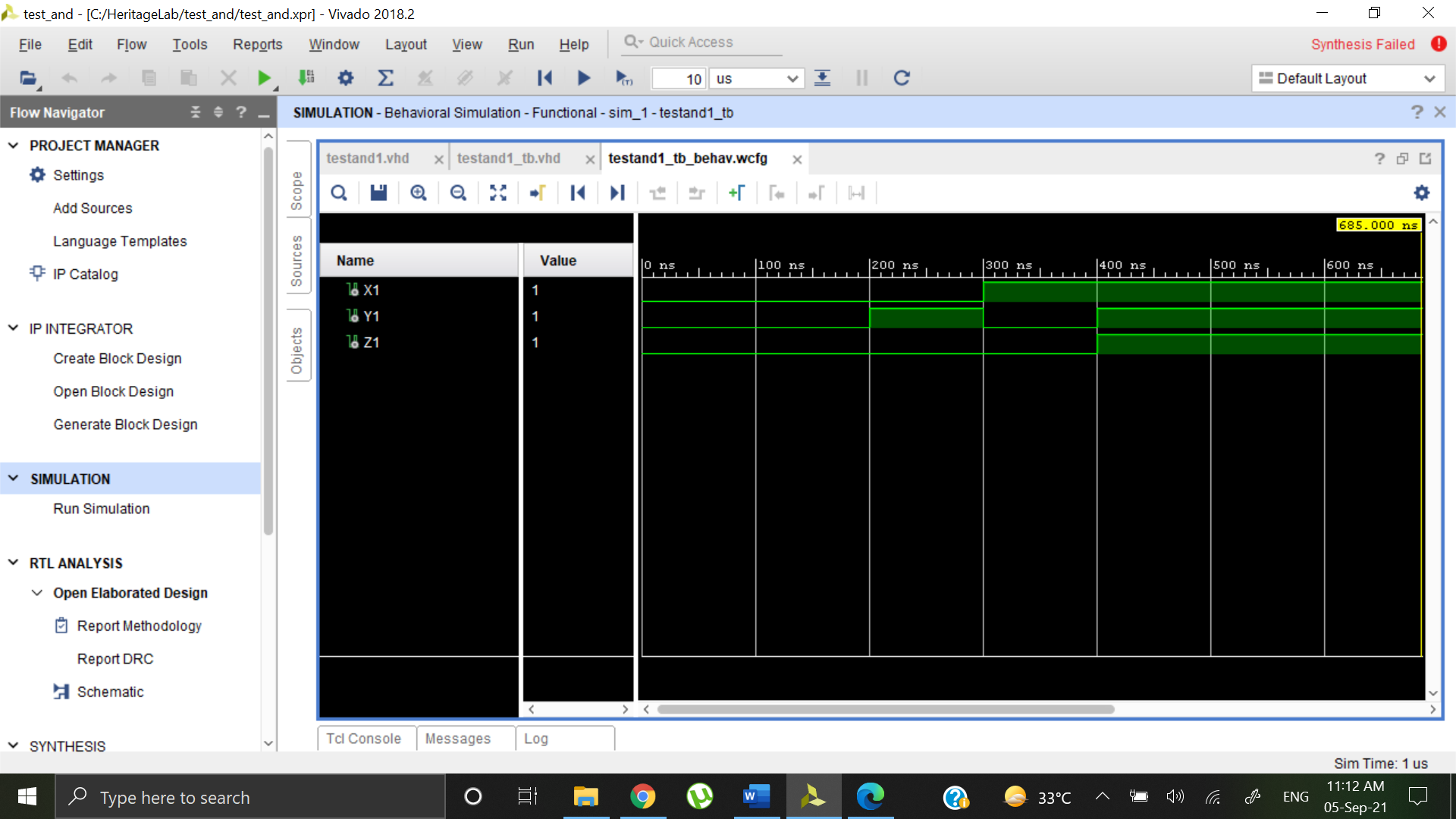
end process;

end Behavioral;

**ELABORATE DESIGN:**



**BEHAVIOURAL SIMULATION:**



**OR GATE**

**TRUTH TABLE:**

|  |  |  |
| --- | --- | --- |
| Input | | Output |
| X | Y | Z |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**DESIGN SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity testor is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end testor;

architecture Behavioral of testor is

begin

Z <= X OR Y;

end Behavioral;

**SIMULATION SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity testor\_tb is

-- Port ( );

end testor\_tb;

architecture Behavioral of testor\_tb is

component testor is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end component;

signal X1 : STD\_LOGIC := '0';

signal Y1 : STD\_LOGIC := '0';

signal Z1 : STD\_LOGIC;

begin

uut: testor port map (X=>X1, Y=>Y1, Z=>Z1);

stim\_proc: process

begin

wait for 100ns;

X1 <= '0';

Y1 <= '0';

wait for 100ns;

X1 <= '0';

Y1 <= '1';

wait for 100ns;

X1 <= '1';

Y1 <= '0';

wait for 100ns;

X1 <= '1';

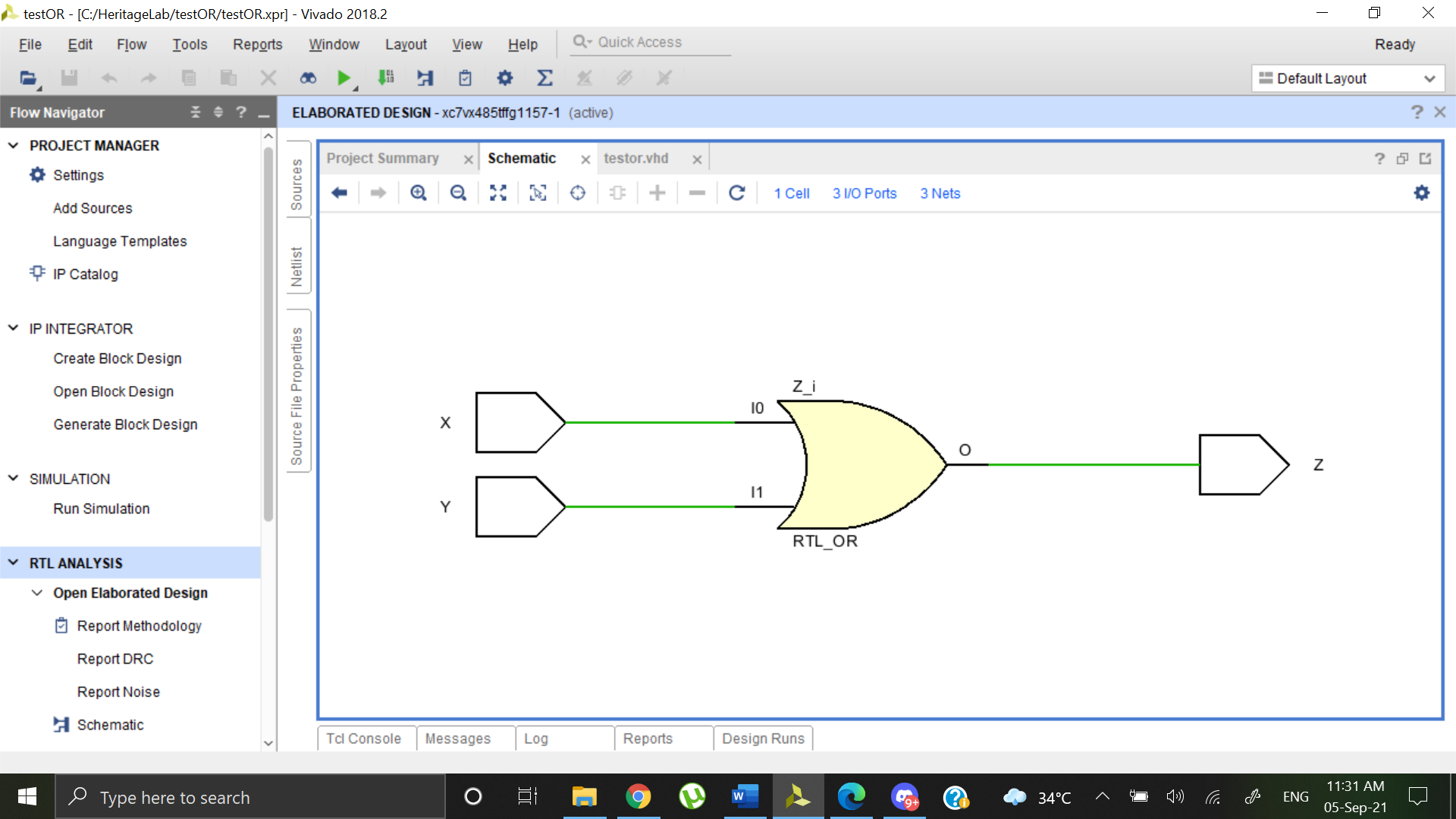
Y1 <= '1';

wait;

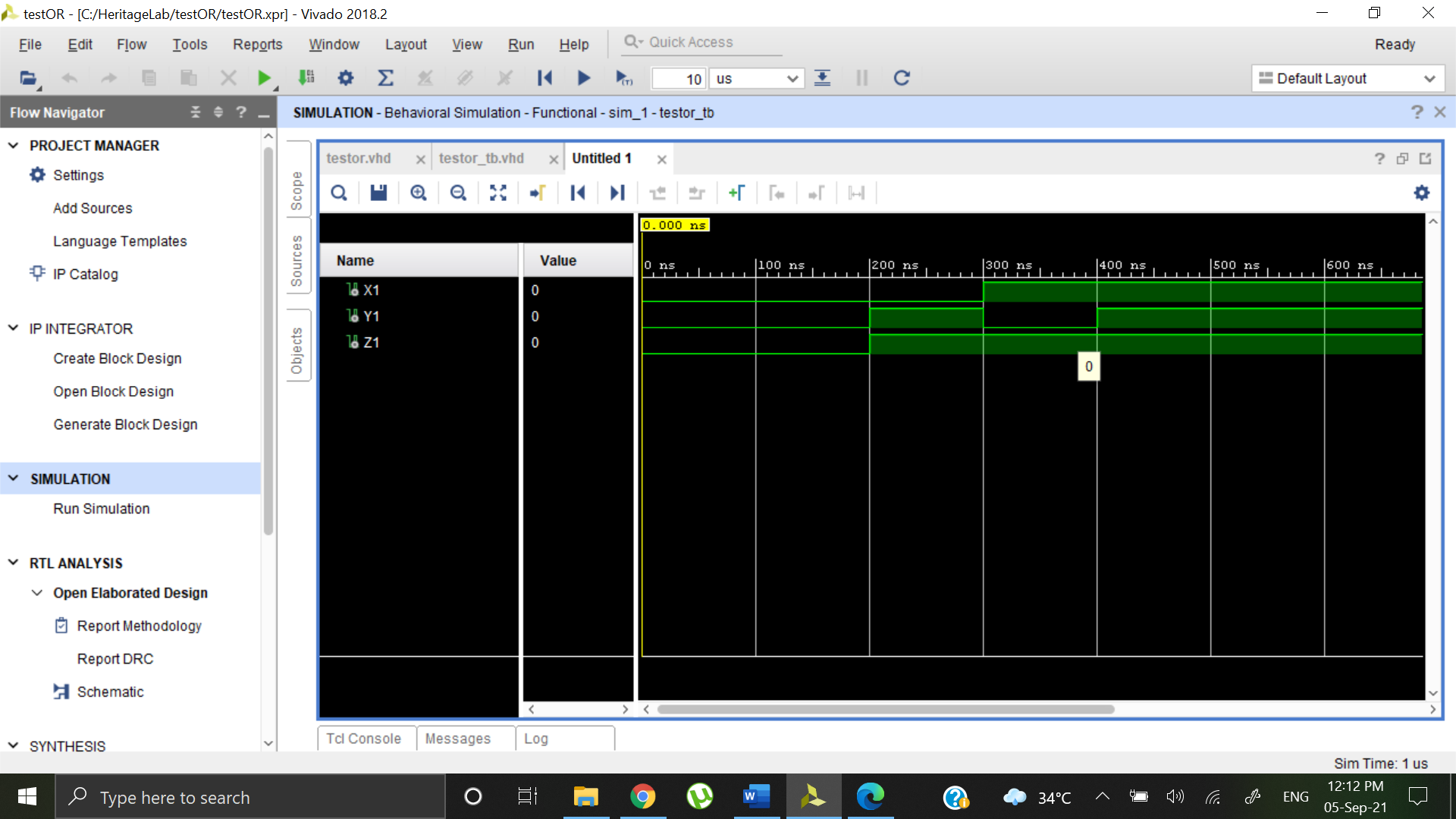
end process;

end Behavioral;

**ELABORATE DESIGN:**



**BEHAVIOURAL SIMULATION:**



**NAND GATE**

**TRUTH TABLE:**

|  |  |  |
| --- | --- | --- |
| Input | | Output |
| X | Y | Z |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**DESIGN SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity testnand is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end testnand;

architecture Dataflow of testnand is

begin

Z<= X NAND Y;

end Dataflow;

**SIMULATION SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity testnand\_tb is

-- Port ( );

end testnand\_tb;

architecture Behavioral of testnand\_tb is

component testnand is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end component;

signal X1 : STD\_LOGIC := '0';

signal Y1 : STD\_LOGIC := '0';

signal Z1 : STD\_LOGIC;

begin

uut: testnand port map (X=>X1, Y=>Y1, Z=>Z1);

stim\_proc: process

begin

wait for 100ns;

X1 <= '0';

Y1 <= '0';

wait for 100ns;

X1 <= '0';

Y1 <= '1';

wait for 100ns;

X1 <= '1';

Y1 <= '0';

wait for 100ns;

X1 <= '1';

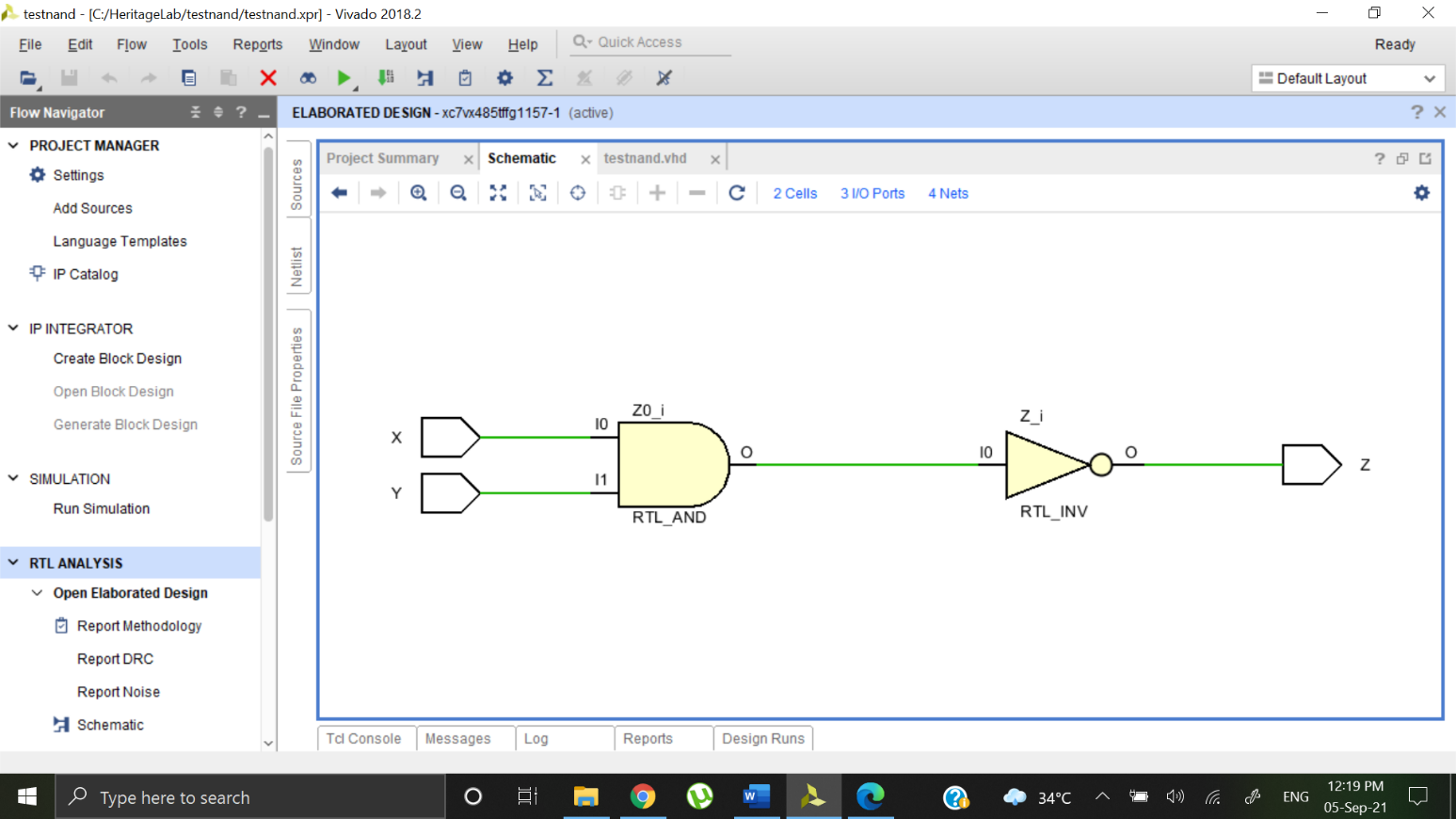
Y1 <= '1';

wait;

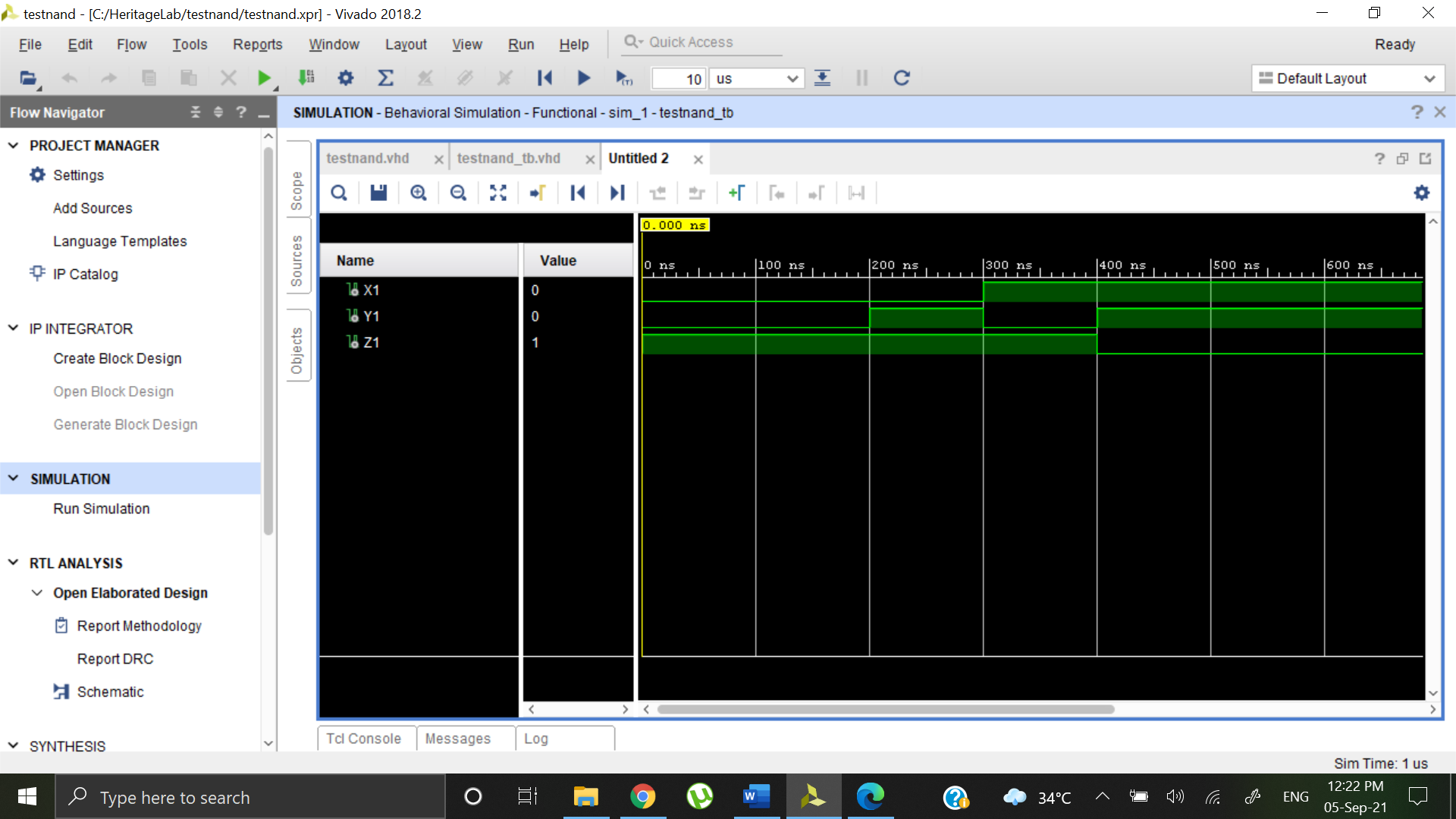
end process;

end Behavioral;

**ELABORATE DESIGN:**



**BEHAVIOURAL SIMULATION:**



**NOR GATE**

**TRUTH TABLE:**

|  |  |  |
| --- | --- | --- |
| Input | | Output |
| X | Y | Z |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

**DESIGN SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity testnor is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end testnor;

architecture Dataflow of testnor is

begin

Z <= X NOR Y;

end Dataflow;

**SIMULATION SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity testnor\_tb is

-- Port ( );

end testnor\_tb;

architecture Behavioral of testnor\_tb is

component testnor is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end component;

signal X1 : STD\_LOGIC := '0';

signal Y1 : STD\_LOGIC := '0';

signal Z1 : STD\_LOGIC;

begin

uut: testnor port map (X=>X1, Y=>Y1, Z=>Z1);

stim\_proc: process

begin

wait for 100ns;

X1 <= '0';

Y1 <= '0';

wait for 100ns;

X1 <= '0';

Y1 <= '1';

wait for 100ns;

X1 <= '1';

Y1 <= '0';

wait for 100ns;

X1 <= '1';

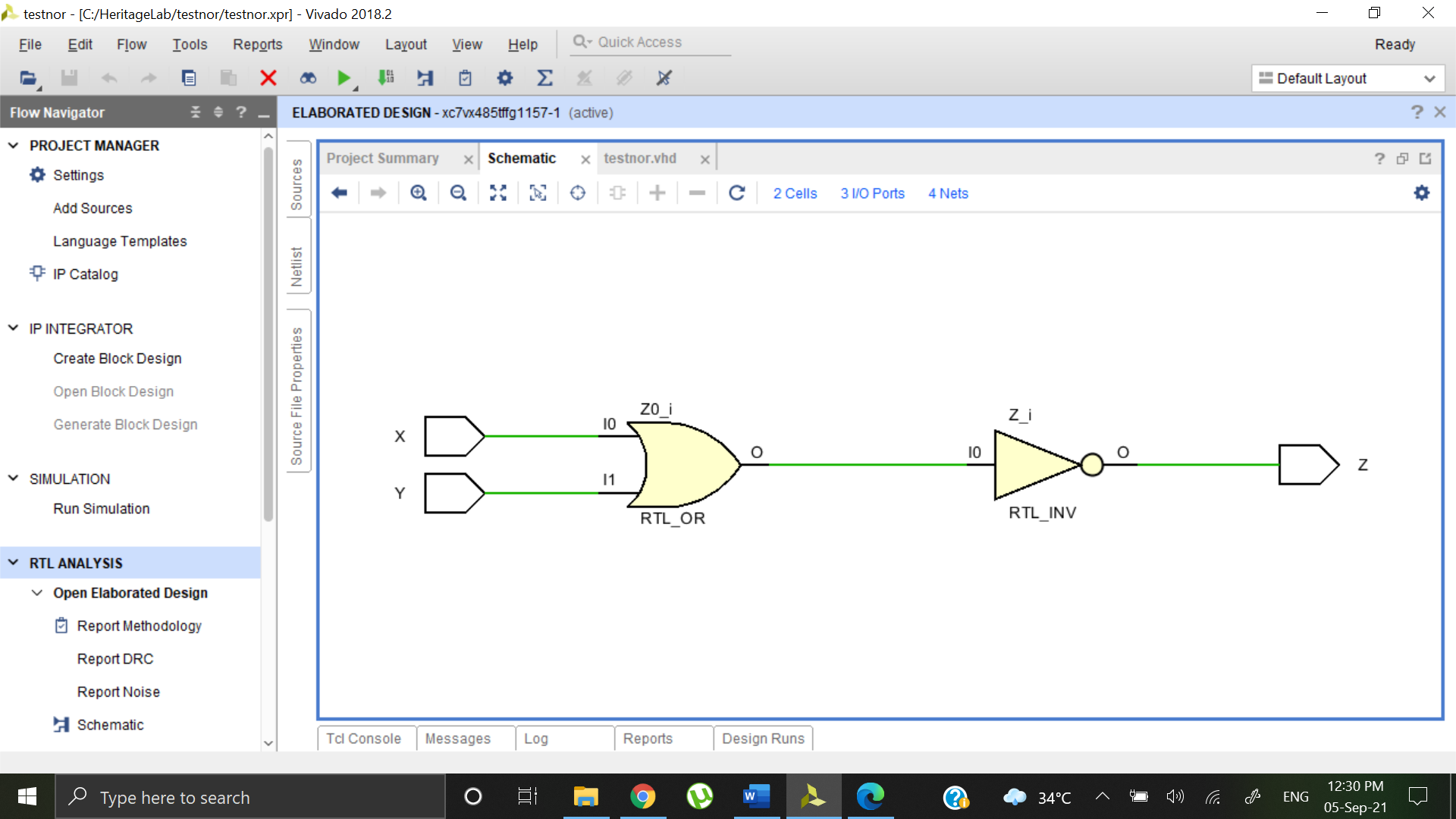
Y1 <= '1';

wait;

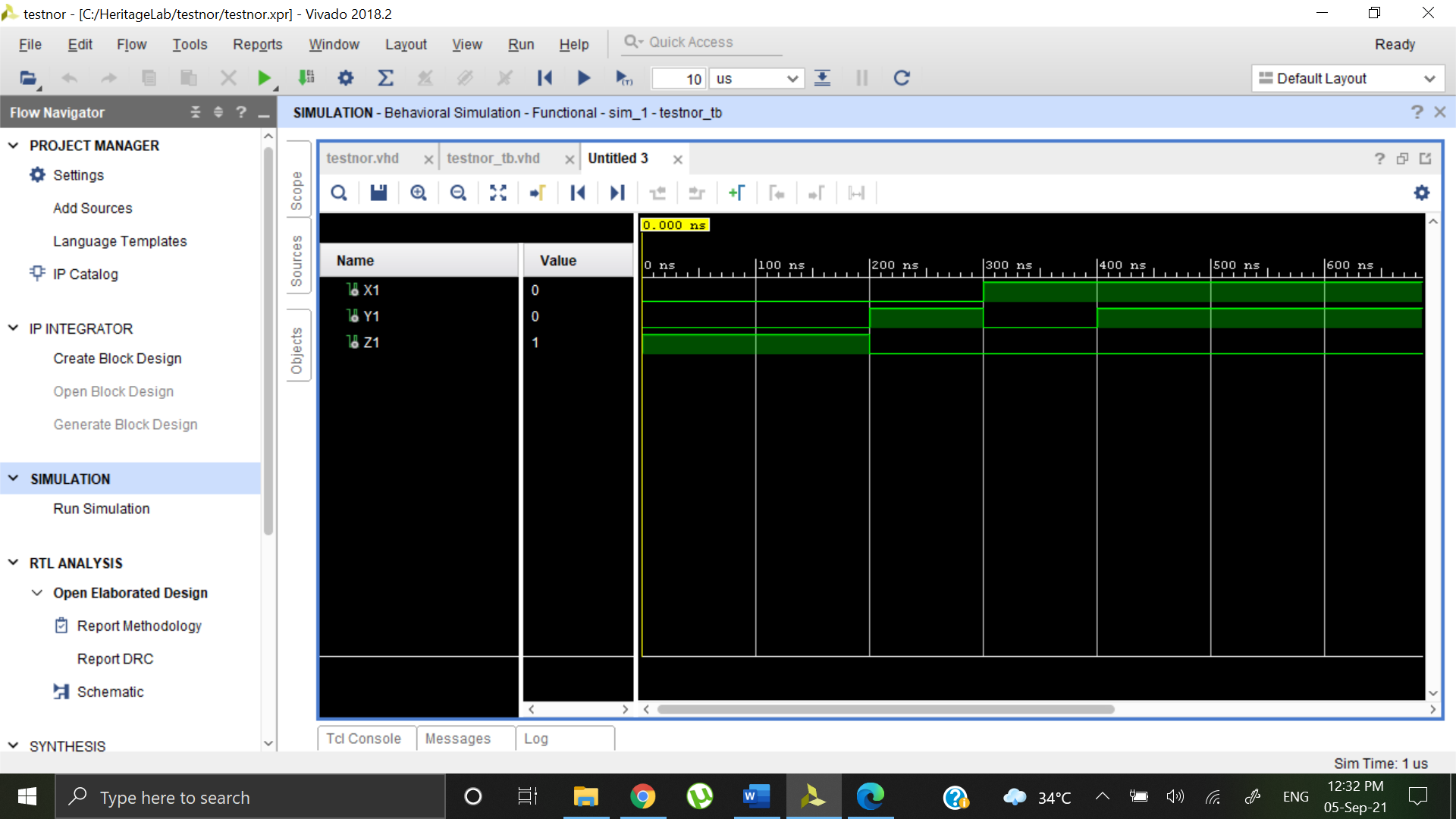
end process;

end Behavioral;

**ELABORATE DESIGN:**



**BEHAVIOURAL SIMULATION:**



**XOR GATE**

**TRUTH TABLE:**

|  |  |  |
| --- | --- | --- |
| Input | | Output |
| X | Y | Z |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**DESIGN SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity testxor is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end testxor;

architecture Dataflow of testnor is

begin

Z <= X XOR Y;

end Dataflow;

**SIMULATION SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity testxor\_tb is

-- Port ( );

end testxor\_tb;

architecture Behavioral of testxor\_tb is

component testxor is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end component;

signal X1 : STD\_LOGIC := '0';

signal Y1 : STD\_LOGIC := '0';

signal Z1 : STD\_LOGIC;

begin

uut: testxor port map (X=>X1, Y=>Y1, Z=>Z1);

stim\_proc: process

begin

wait for 100ns;

X1 <= '0';

Y1 <= '0';

wait for 100ns;

X1 <= '0';

Y1 <= '1';

wait for 100ns;

X1 <= '1';

Y1 <= '0';

wait for 100ns;

X1 <= '1';

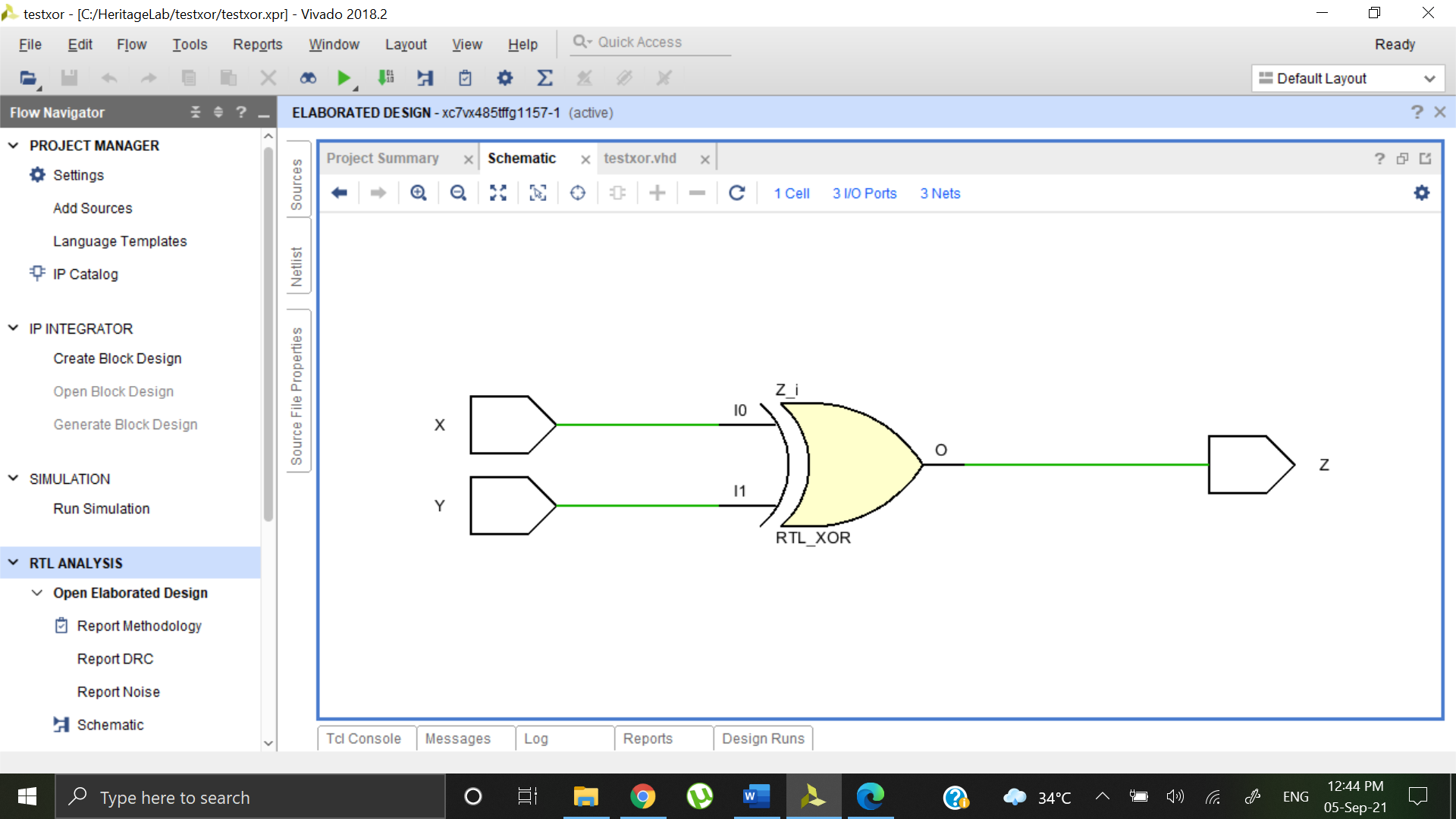
Y1 <= '1';

wait;

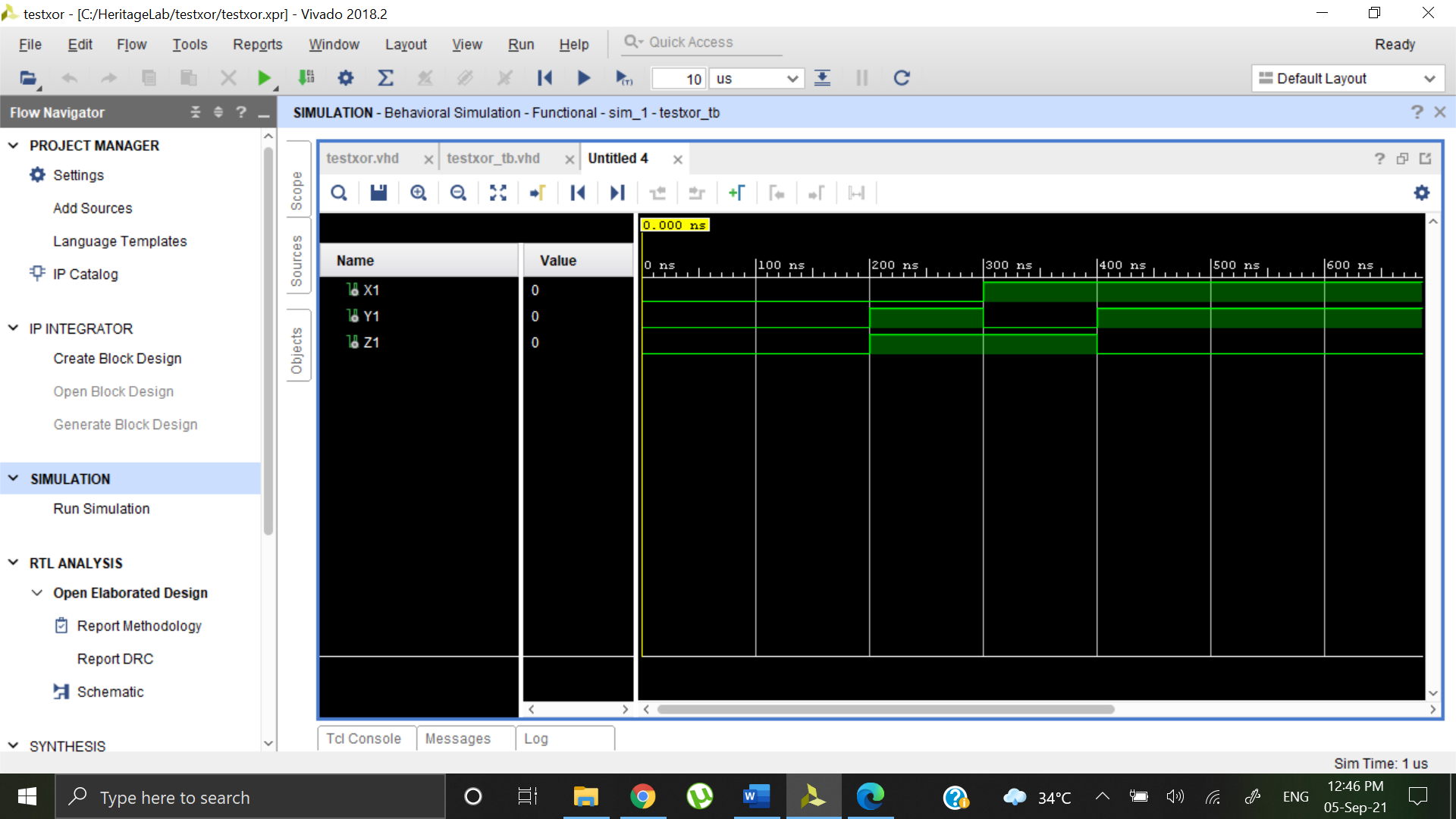
end process;

end Behavioral;

**ELABORATE DESIGN:**



**SIMULATION SOURCE:**



**NOT GATE**

**TRUTH TABLE:**

|  |  |
| --- | --- |
| Input | Output |
| X | Z |
| 0 | 1 |
| 1 | 0 |

**DESIGN SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity testnot is

Port ( X : in STD\_LOGIC;

Z : out STD\_LOGIC);

end testnot;

architecture Dataflow of testnot is

begin

Z <= NOT X;

end Dataflow;

**SIMULATION SOURCE:**

architecture Dataflow of testnot is

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity testnot\_tb is

-- Port ( );

end testnot\_tb;

architecture Behavioral of testnot\_tb is

component testnot is

Port ( X : in STD\_LOGIC;

Z : out STD\_LOGIC);

end component;

signal X1 : STD\_LOGIC := '0';

signal Z1 : STD\_LOGIC;

begin

uut: testnot port map (X=>X1, Z=>Z1);

stim\_proc: process

begin

wait for 100ns;

X1 <= '0';

wait for 100ns;

X1 <= '1';

wait;

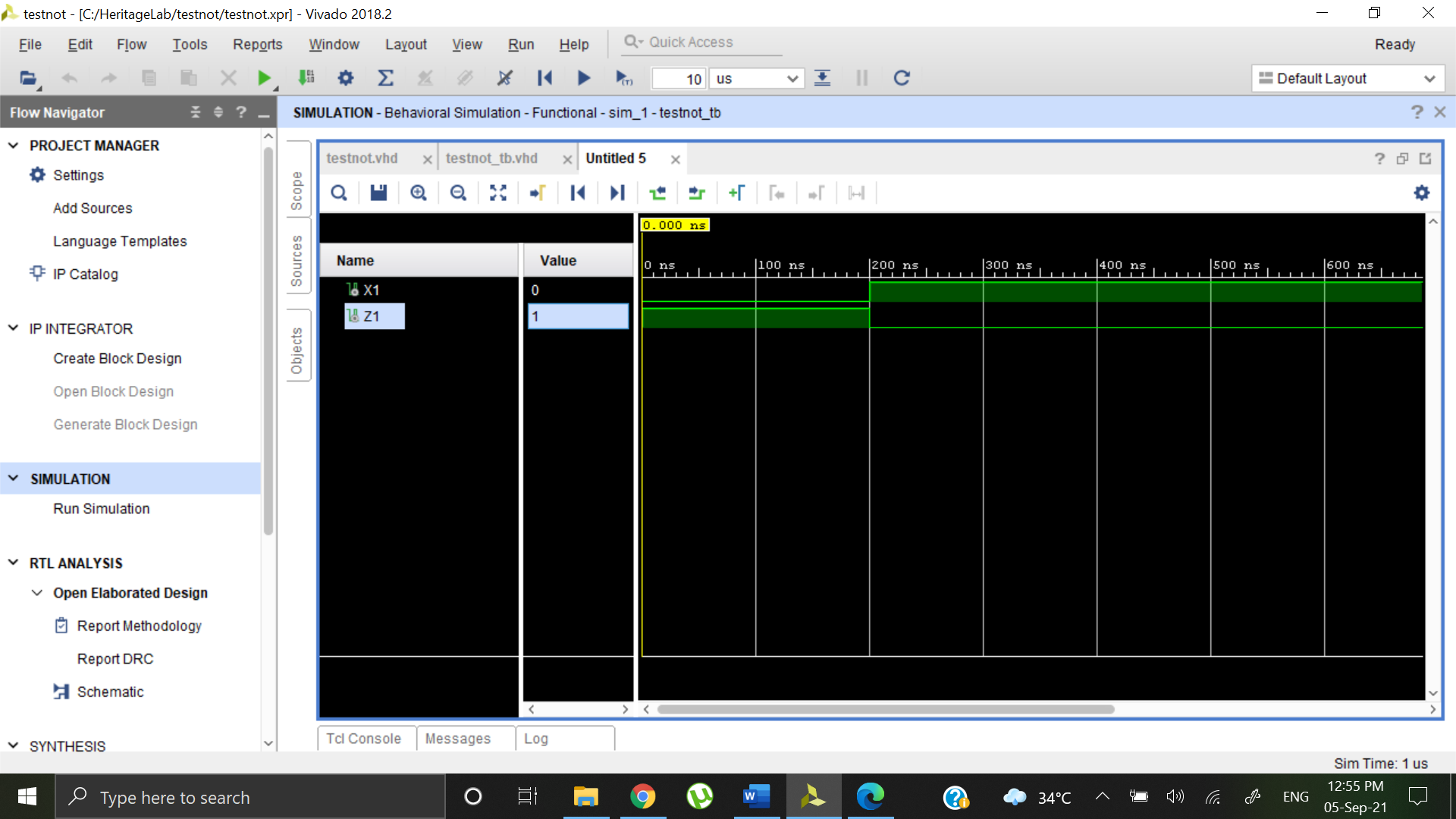
end process;

end Behavioral;

**ELABORATE DESIGN:**



**SIMULATION SOURCE:**



**AND using NAND**

**DESIGN SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity testOR1 is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end testOR1;

architecture Dataflow of testOR1 is

begin

Z <= (X NAND Y)NAND (X NAND Y);

end Dataflow;

**SIMULATION SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity testand1\_tb is

-- Port ( );

end testand1\_tb;

architecture Behavioral of testand1\_tb is

component testOR1 is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end component;

signal X1 : STD\_LOGIC := '0';

signal Y1 : STD\_LOGIC := '0';

signal Z1 : STD\_LOGIC;

begin

uut: testOR1 port map (X=>X1, Y=>Y1, Z=>Z1);

stim\_proc: process

begin

wait for 100ns;

X1 <= '0';

Y1 <= '0';

wait for 100ns;

X1 <= '0';

Y1 <= '1';

wait for 100ns;

X1 <= '1';

Y1 <= '0';

wait for 100ns;

X1 <= '1';

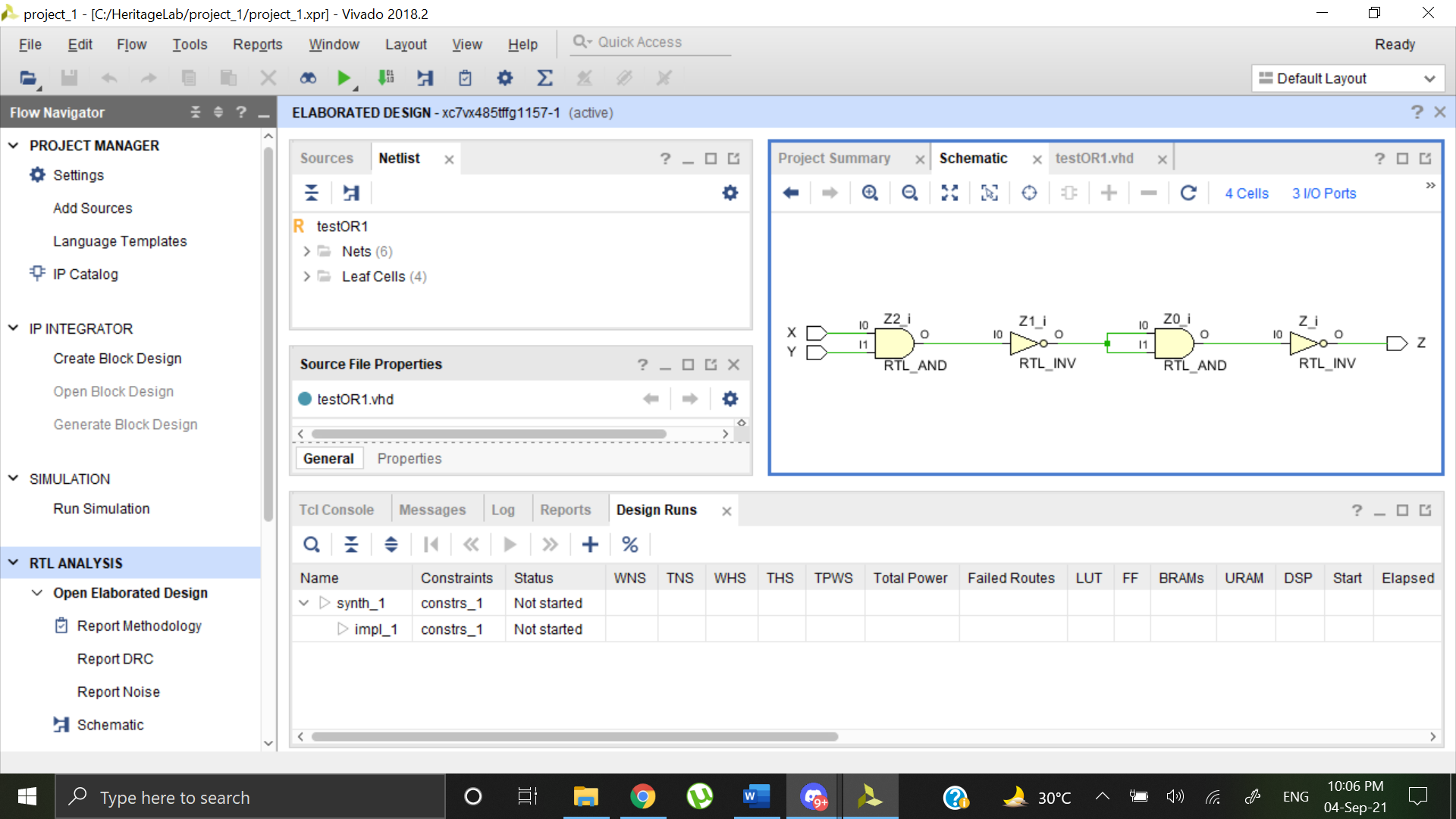
Y1 <= '1';

wait;

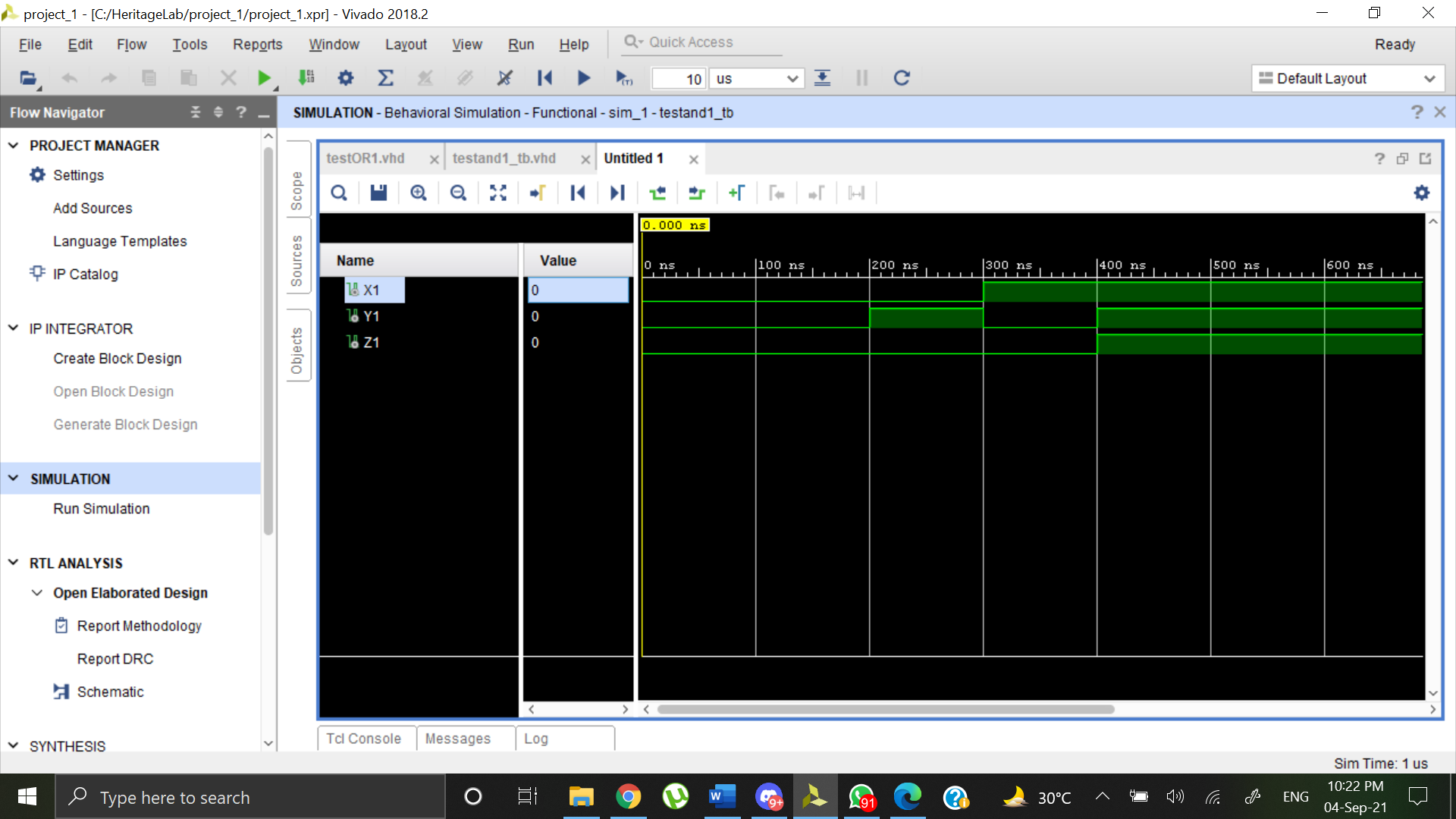
end process;

end Behavioral;

**ELABORATE DESIGN:**



**BEHAVIORAL SIMULATION:**



**OR using NAND**

**DESIGN SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity testOR1 is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end testOR1;

architecture Dataflow of testOR1 is

begin

Z <= (X NAND Y)NAND (X NAND Y);

end Dataflow;

**SIMULATION SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity ornand\_tb is

-- Port ( );

end ornand\_tb;

architecture Behavioral of ornand\_tb is

component ornand is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end component;

signal X1 : STD\_LOGIC := '0';

signal Y1 : STD\_LOGIC := '0';

signal Z1 : STD\_LOGIC;

begin

uut: ornand port map (X=>X1, Y=>Y1, Z=>Z1);

stim\_proc: process

begin

wait for 100ns;

X1 <= '0';

Y1 <= '0';

wait for 100ns;

X1 <= '0';

Y1 <= '1';

wait for 100ns;

X1 <= '1';

Y1 <= '0';

wait for 100ns;

X1 <= '1';

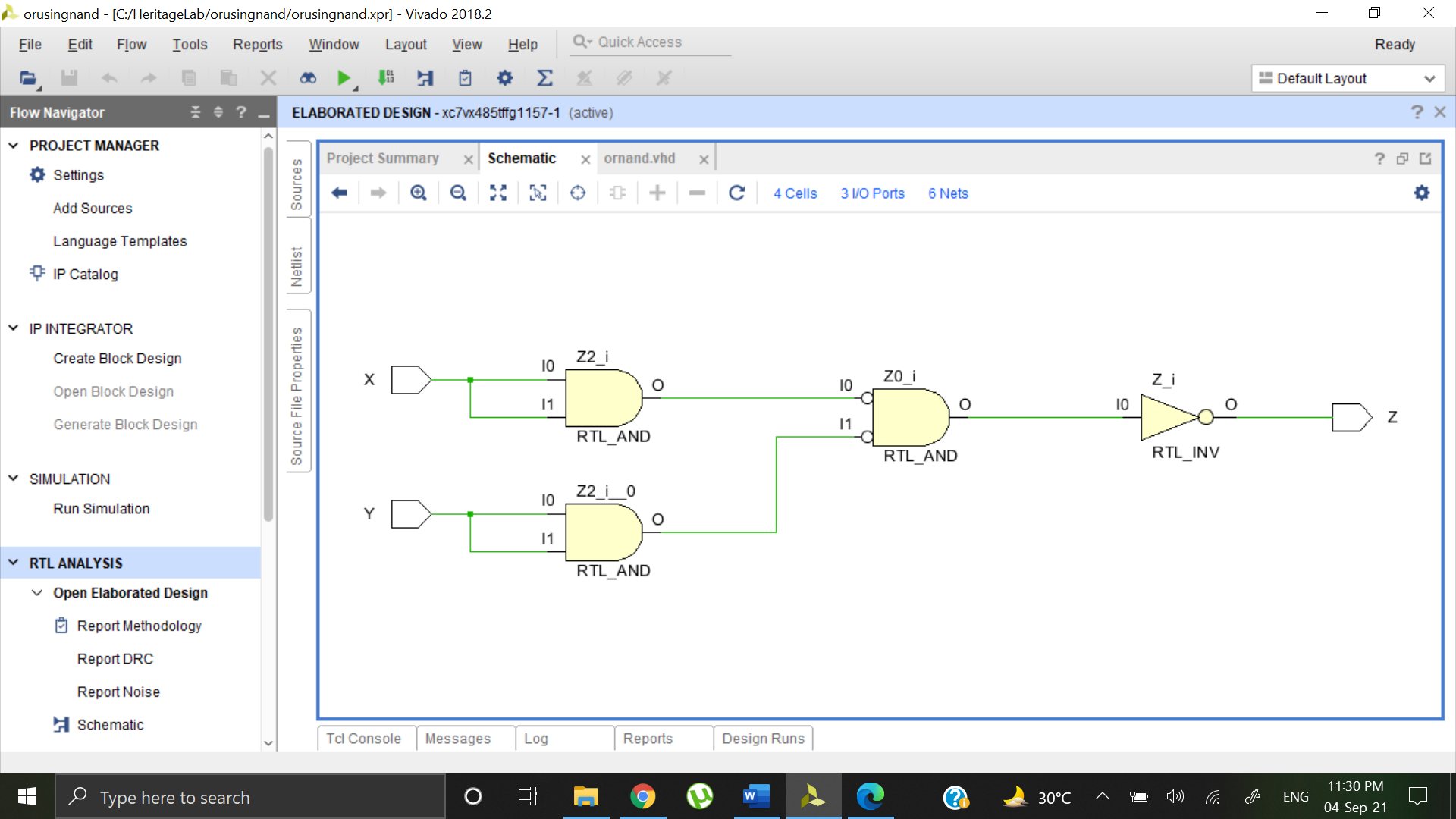
Y1 <= '1';

wait;

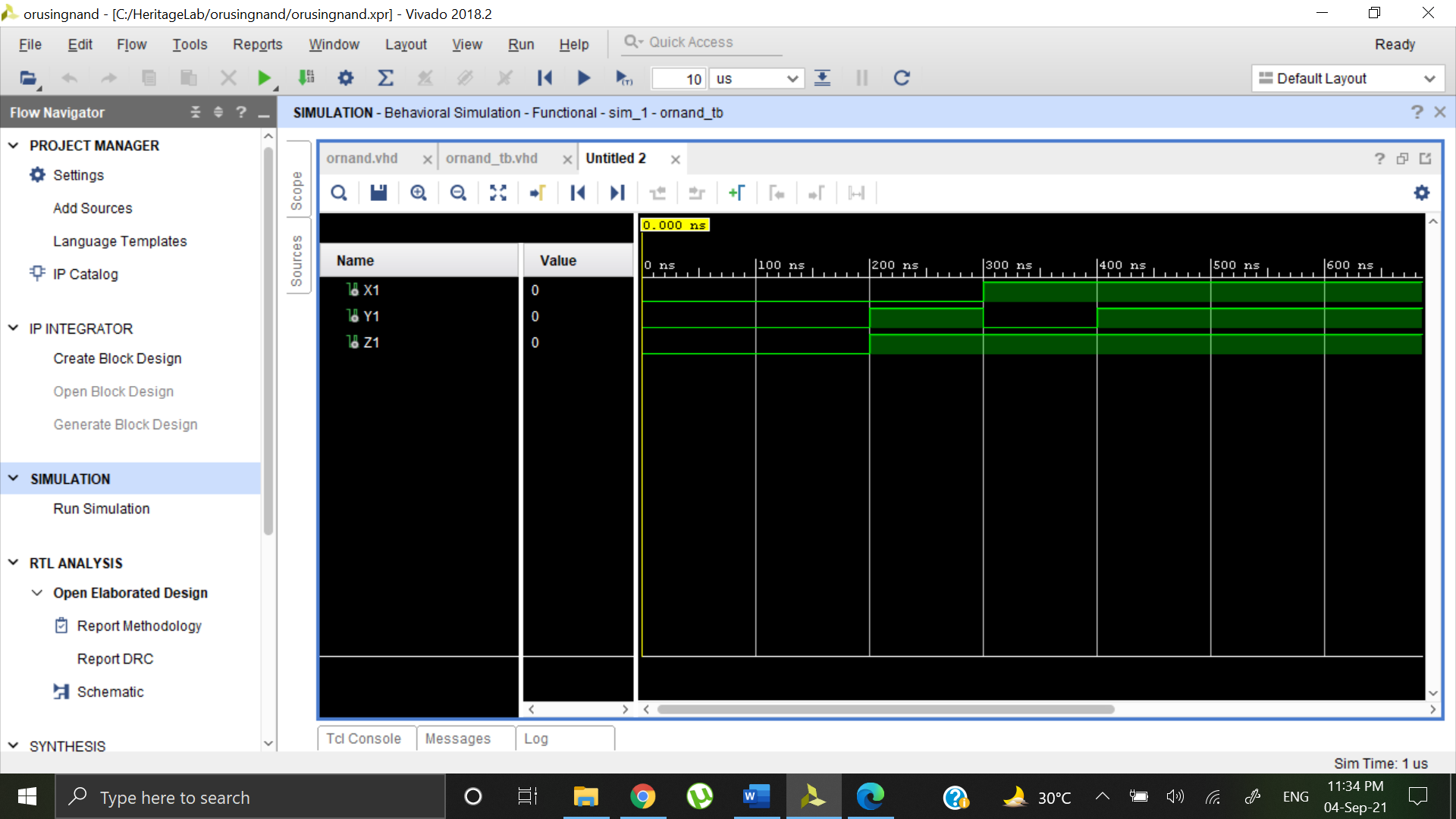
end process;

end Behavioral;

**ELABORATE DESIGN:**



**BEHAVIORAL SIMULATION:**



**NOT using NAND**

**DESIGN SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity testOR1 is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end testOR1;

architecture Dataflow of testOR1 is

begin

Z <= X NAND X;

end Dataflow;

**SIMULATION SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity notnand\_tb is

-- Port ( );

end notnand\_tb;

architecture Behavioral of notnand\_tb is

component notnand is

Port ( X : in STD\_LOGIC;

Y: in STD\_LOGIC;

Z : out STD\_LOGIC);

end component;

signal X1 : STD\_LOGIC := '0';

signal Y1: STD\_LOGIC;

signal Z1 : STD\_LOGIC;

begin

uut: notnand port map (X=>X1,Y =>Y1,Z=>Z1);

stim\_proc: process

begin

wait for 100ns;

X1 <= '0';

wait for 100ns;

X1 <= '0';

wait for 100ns;

X1 <= '1';

wait for 100ns;

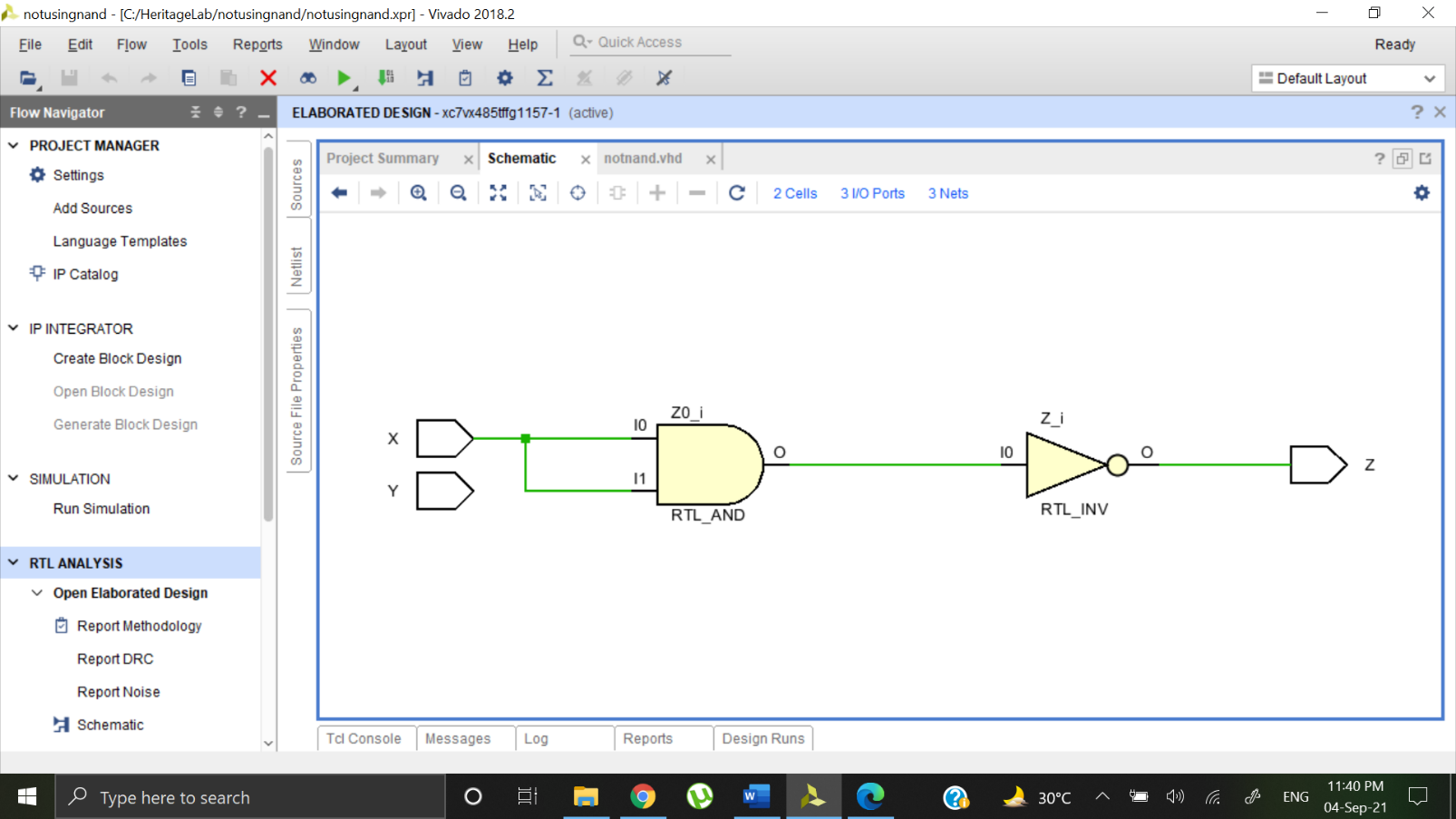
X1 <= '1';

wait;

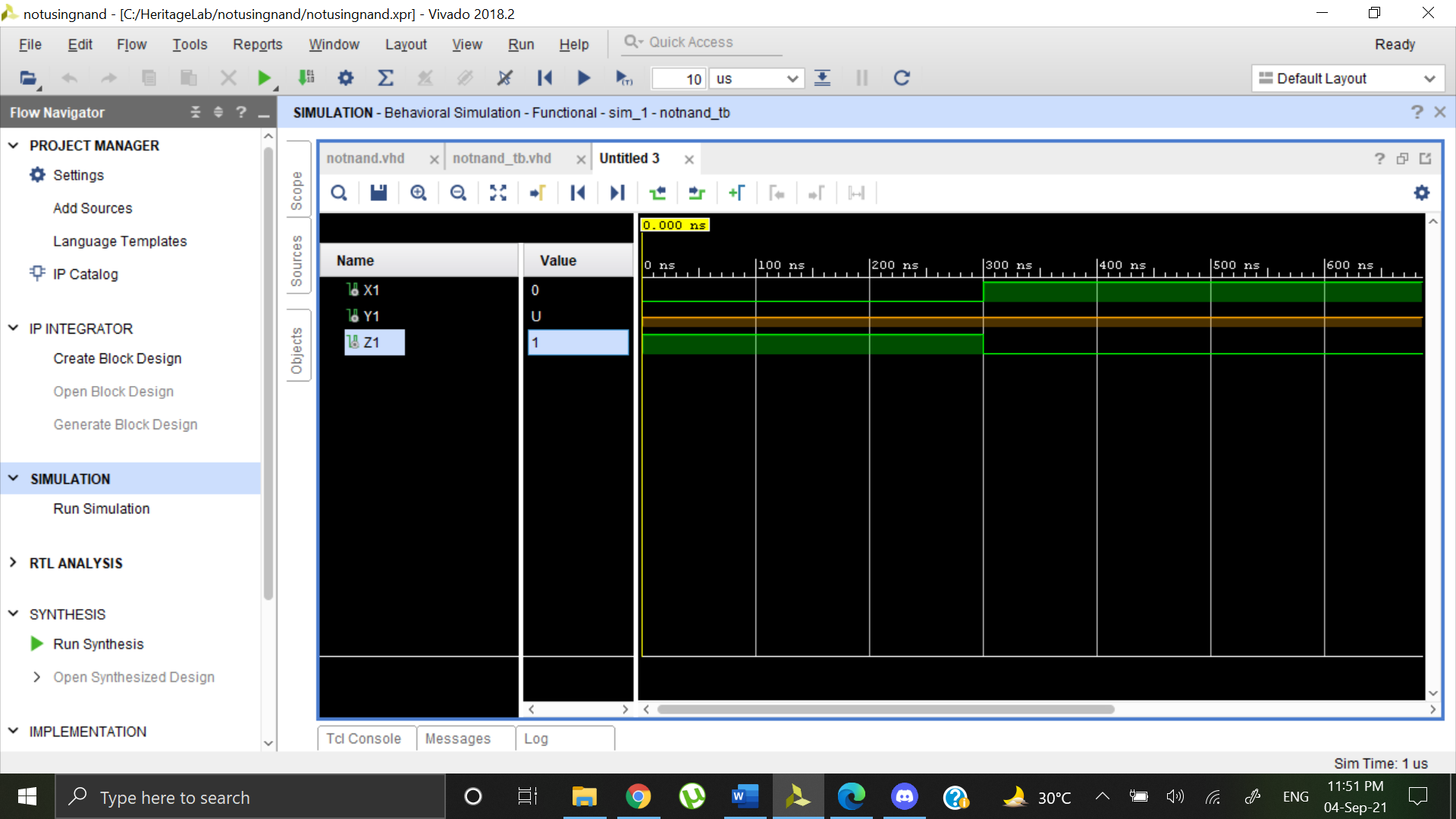
end process;

end Behavioral;

**ELABORATE DESIGN:**



**BEHAVIORAL SIMULATION:**



**AND using NOR**

**DESIGN SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity andnor is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end andnor;

architecture Dataflow of andnor is

begin

Z <= (X NOR X) NOR (Y NOR Y);

end Dataflow;

**SIMULATION SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity andnor\_tb is

-- Port ( );

end andnor\_tb;

architecture Behavioral of andnor\_tb is

component andnor is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end component;

signal X1 : STD\_LOGIC := '0';

signal Y1 : STD\_LOGIC := '0';

signal Z1 : STD\_LOGIC;

begin

uut: andnor port map (X=>X1, Y=>Y1, Z=>Z1);

stim\_proc: process

begin

wait for 100ns;

X1 <= '0';

Y1 <= '0';

wait for 100ns;

X1 <= '0';

Y1 <= '1';

wait for 100ns;

X1 <= '1';

Y1 <= '0';

wait for 100ns;

X1 <= '1';

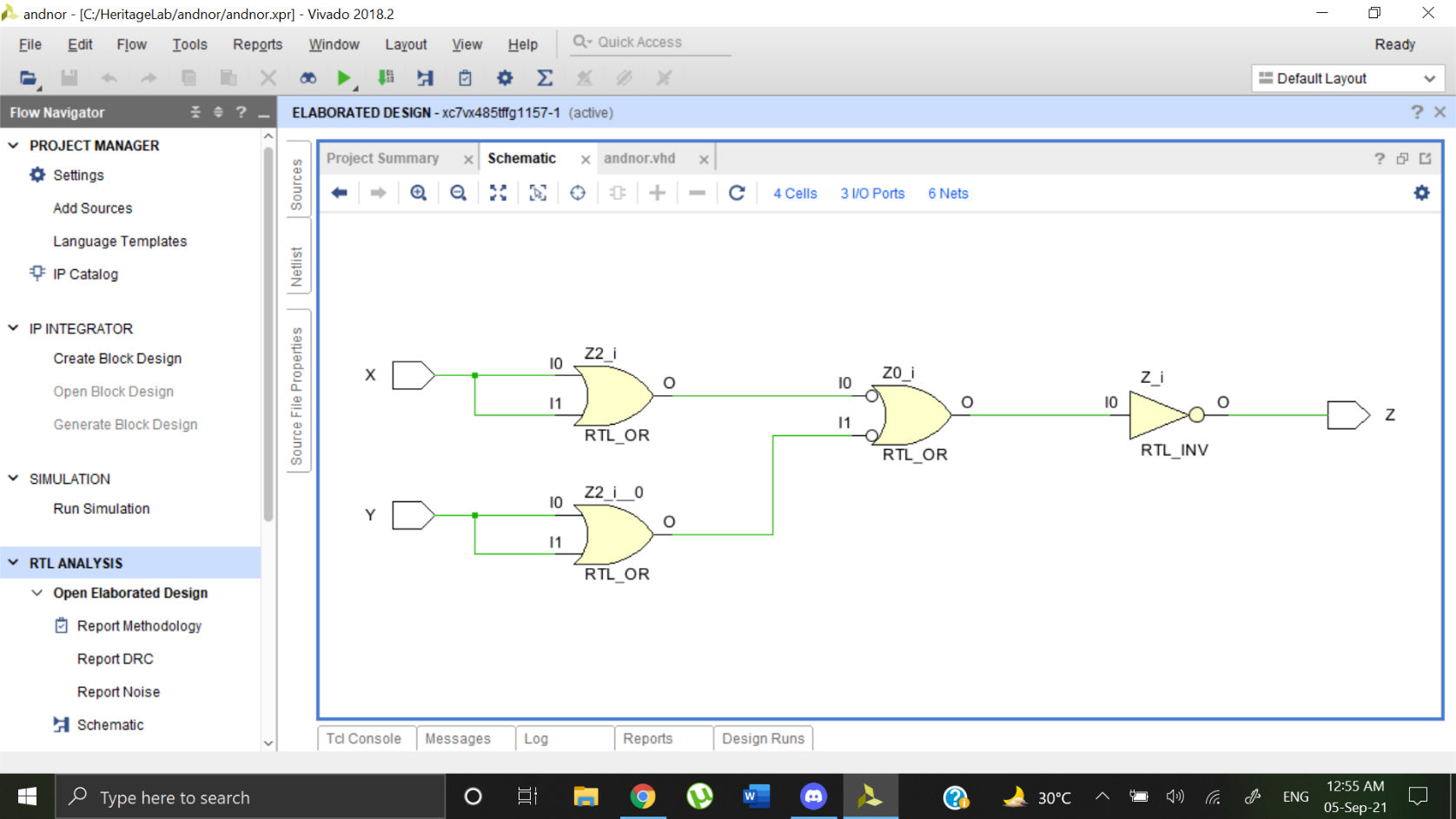
Y1 <= '1';

wait;

end process;

end Behavioral;

**ELABORATE DESIGN:**



**BEHAVIORAL SIMULATION:**



**OR using NOR**

**DESIGN SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity ornor is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end ornor;

architecture Dataflow of ornor is

begin

Z <= (X NOR Y) NOR (X NOR Y);

end Dataflow;

**SIMULATION SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity ornor\_tb is

-- Port ( );

end ornor\_tb;

architecture Behavioral of ornor\_tb is

component ornor is

Port ( X : in STD\_LOGIC;

Y : in STD\_LOGIC;

Z : out STD\_LOGIC);

end component;

signal X1 : STD\_LOGIC := '0';

signal Y1 : STD\_LOGIC := '0';

signal Z1 : STD\_LOGIC;

begin

uut: ornor port map (X=>X1, Y=>Y1, Z=>Z1);

stim\_proc: process

begin

wait for 100ns;

X1 <= '0';

Y1 <= '0';

wait for 100ns;

X1 <= '0';

Y1 <= '1';

wait for 100ns;

X1 <= '1';

Y1 <= '0';

wait for 100ns;

X1 <= '1';

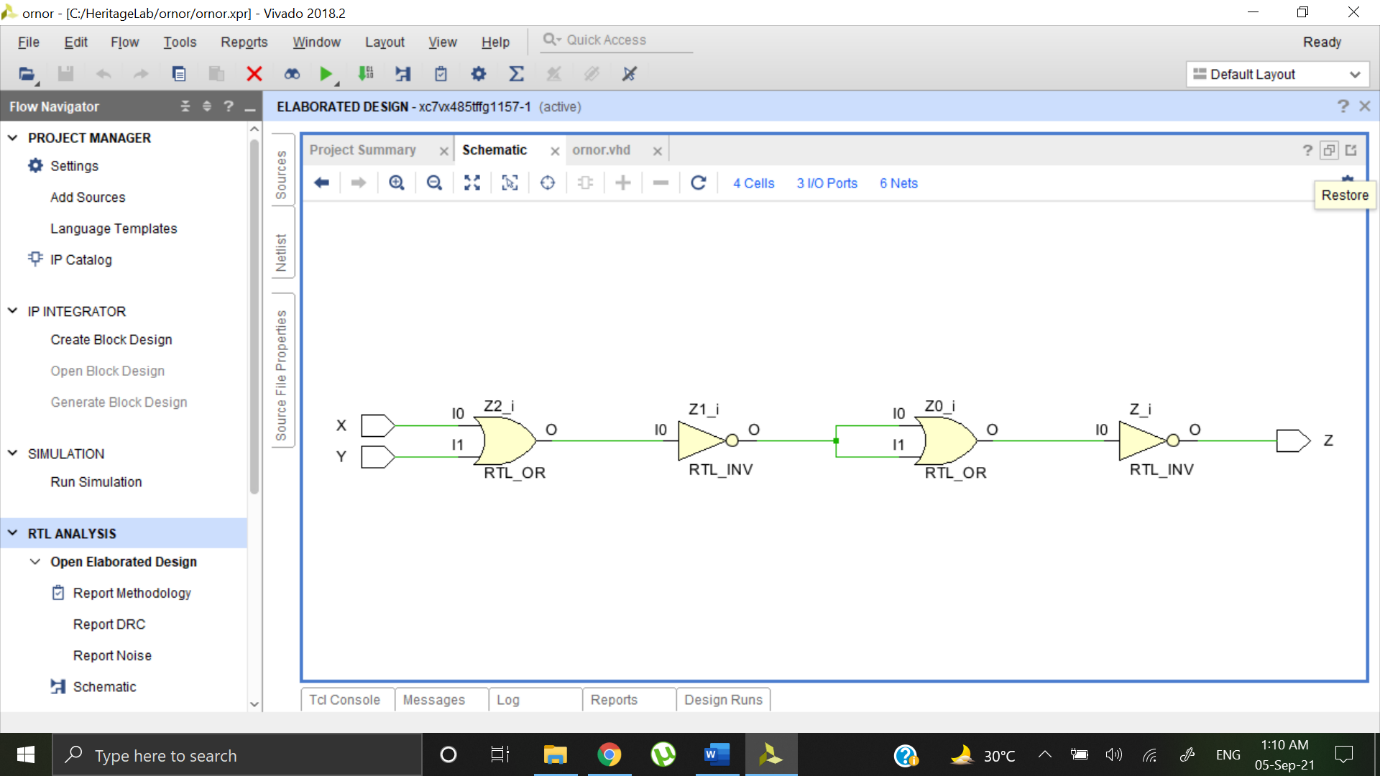
Y1 <= '1';

wait;

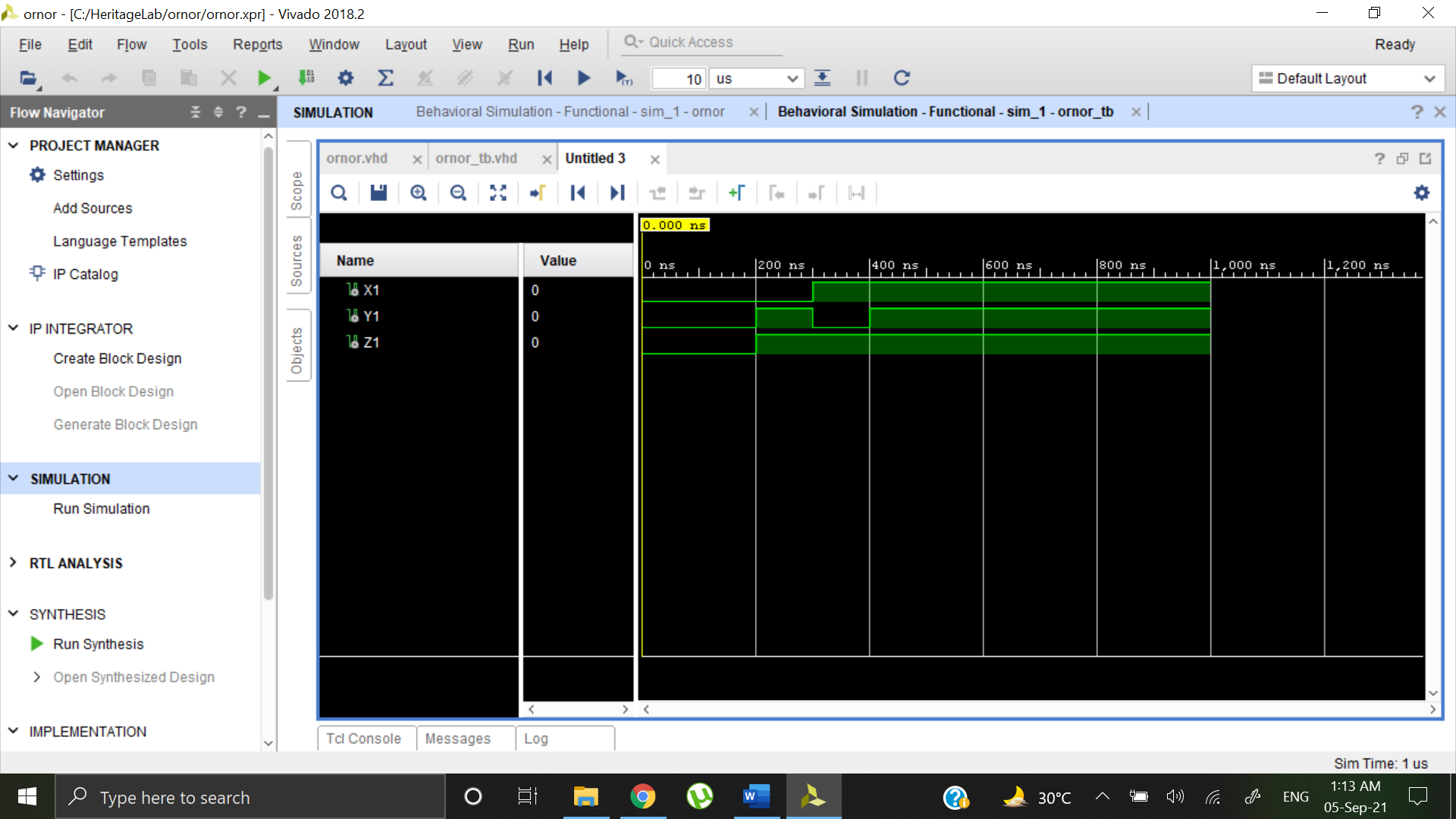
end process;

end Behavioral;

**ELABORATE DESIGN:**



**BEHAVIORAL SIMULATION:**



**NOT using NOR**

**DESIGN SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity notnor is

Port ( X : in STD\_LOGIC;

Z : out STD\_LOGIC);

end notnor;

architecture Dataflow of notnor is

begin

Z <= X NOR X;

end Dataflow;

**SIMULATION SOURCE:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity notnor\_tb is

-- Port ( );

end notnor\_tb;

architecture Behavioral of notnor\_tb is

component notnor is

Port ( X : in STD\_LOGIC;

Z : out STD\_LOGIC);

end component;

signal X1 : STD\_LOGIC := '0';

signal Z1 : STD\_LOGIC;

begin

uut: notnor port map (X=>X1, Z=>Z1);

stim\_proc: process

begin

wait for 100ns;

X1 <= '0';

wait for 100ns;

X1 <= '0';

wait for 100ns;

X1 <= '1';

wait for 100ns;

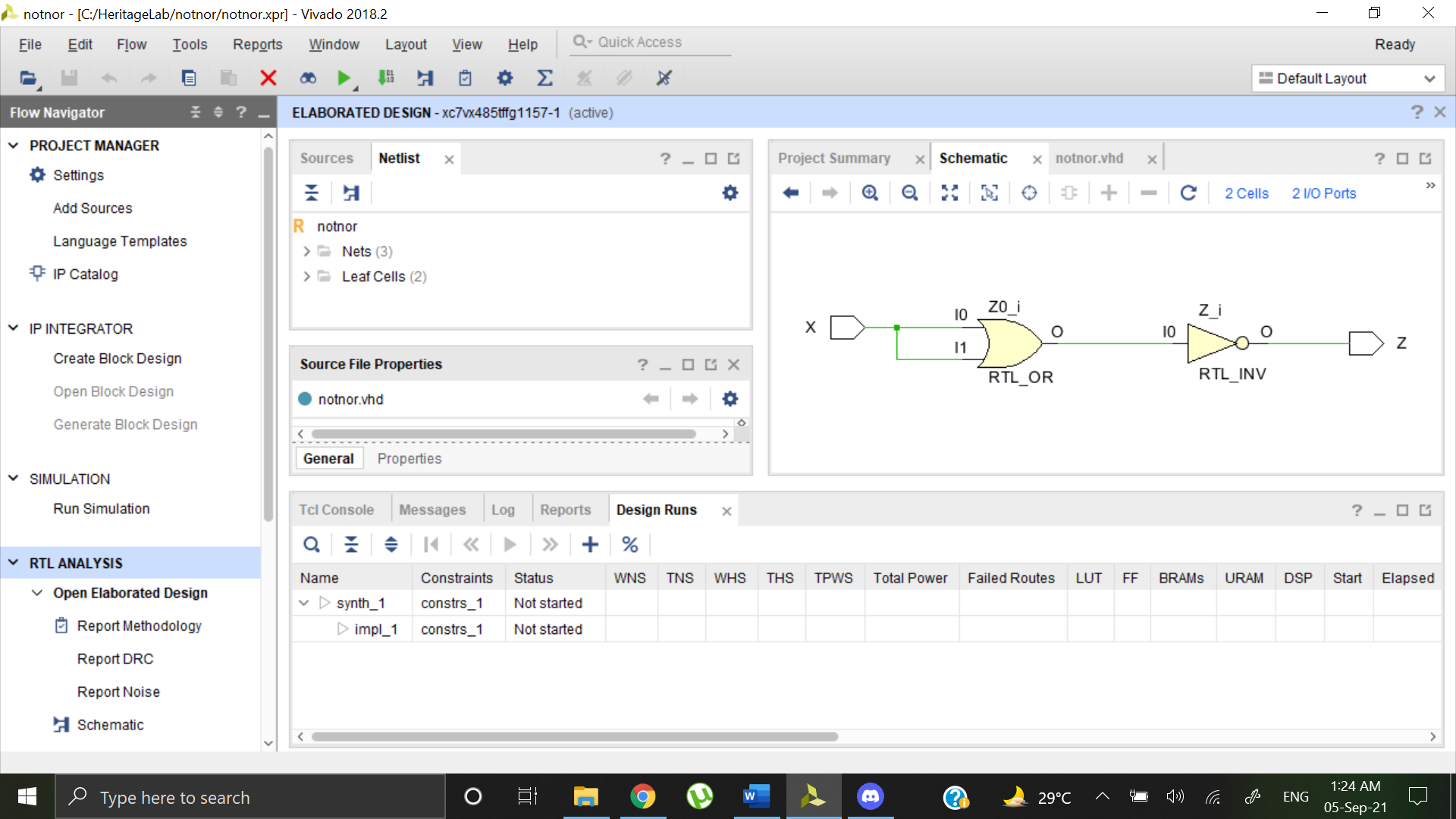
X1 <= '1';

wait;

end process;

end Behavioral;

**ELABORATE DESIGN:**



**BEHAVIORAL SIMULATION:**

