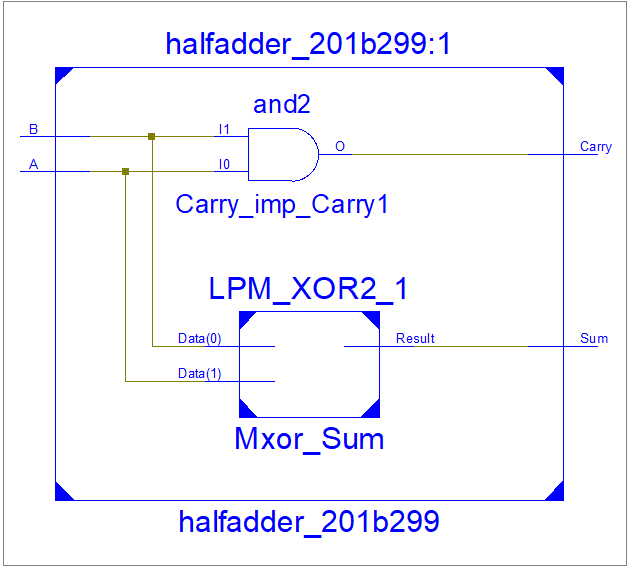
# Experiment 2

**Aim: Design of basic binary adders and subtractors.**

**Exercise 1:** Design half adder in **data flow style** of modeling.

**Design code:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity halfadder\_201b299 is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

Sum : out STD\_LOGIC;

Carry : out STD\_LOGIC);

end halfadder\_201b299;

architecture Dataflow of halfadder\_201b299 is

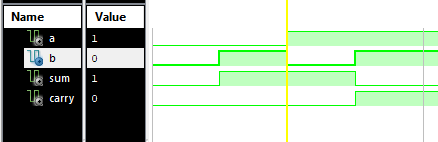
begin

Sum <= A xor B;

Carry <= A and B;

end Dataflow;

**Test Bench code:**

A <= '0'; B<= '0';

wait for 100 ns;

A <= '0'; B<= '1';

wait for 100 ns;

A <= '1'; B<= '0';

wait for 100 ns;

A <= '1'; B<= '1';

wait for 100 ns;

**Exercise#2:** Design half subtractor in **behavioral style** (using either **if-then** or **case when**) of

modeling.

**Design code:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity HalfSubtractor\_201b299 is

Port ( A : in STD\_LOGIC;

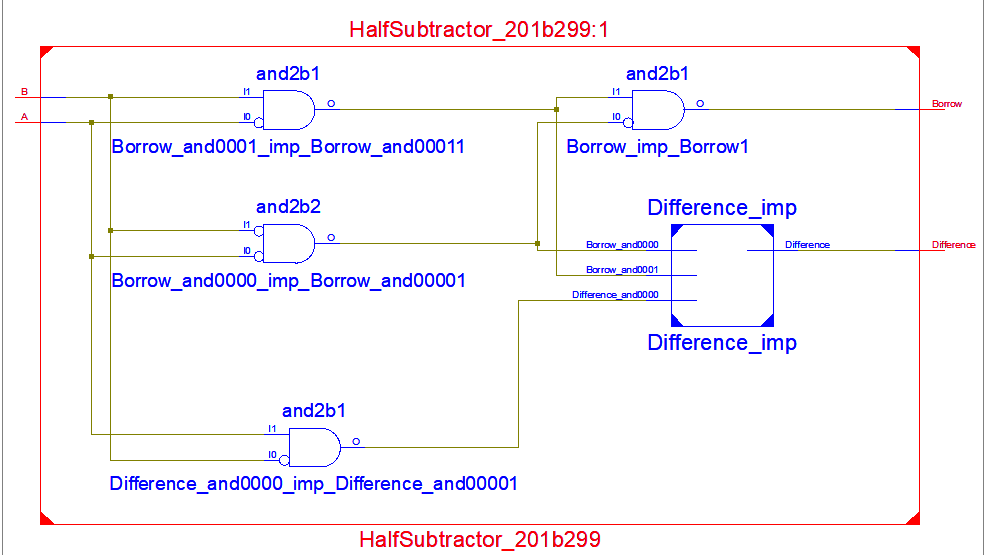
B : in STD\_LOGIC;

Difference : out STD\_LOGIC;

Borrow : out STD\_LOGIC);

end HalfSubtractor\_201b299;

architecture Behavioral of HalfSubtractor\_201b299 is



begin

process(A,B)

begin

if(A = '0' and B = '0')then

Difference <= '0';

Borrow <= '0';

elsif(A = '0' and B = '1')then

Difference <= '1';

Borrow <= '1';

elsif(A = '1' and B = '0')then

Difference <= '1';

Borrow <= '0';

else

Difference <= '0';

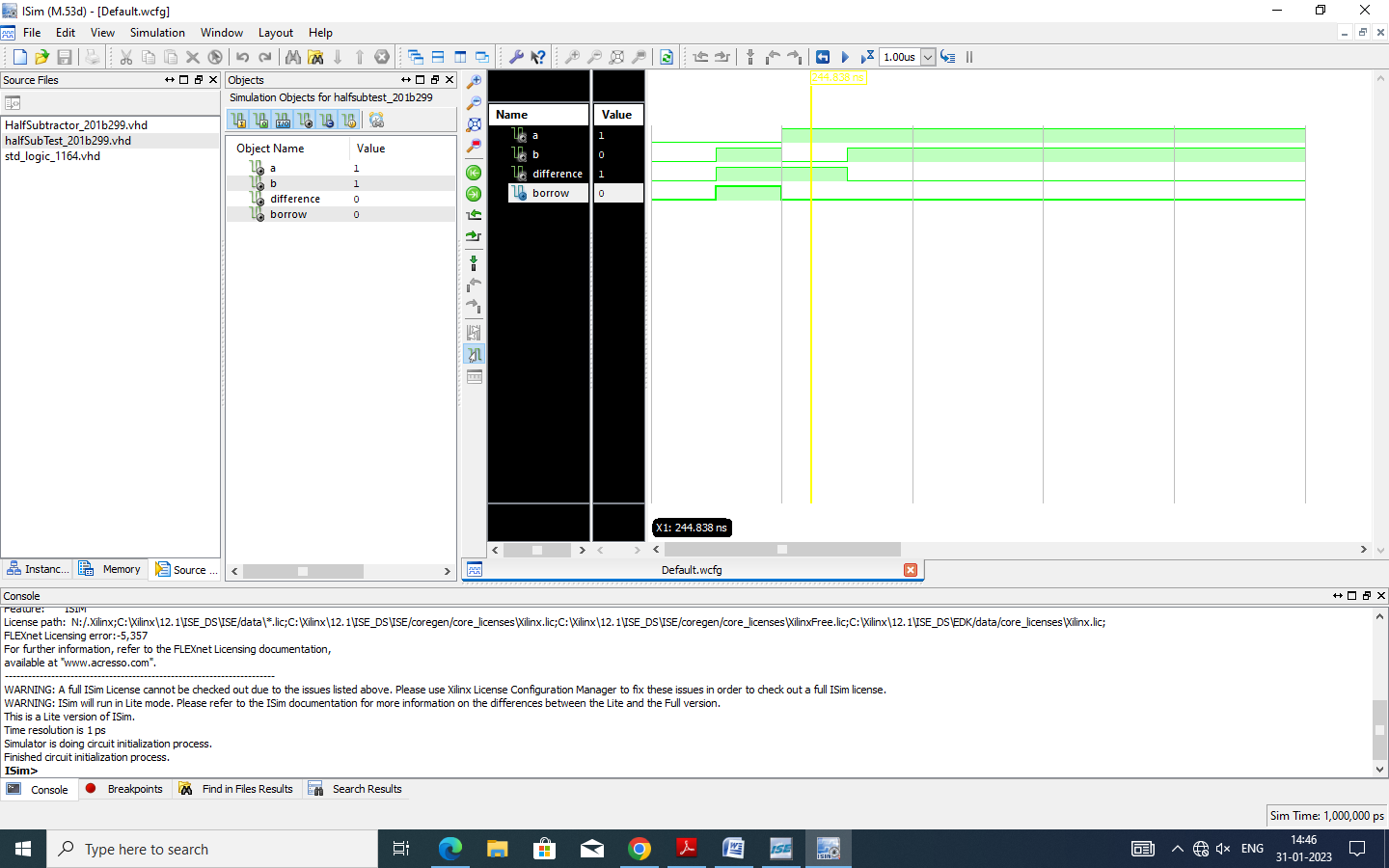
Borrow <= '0';

end if;

end process;

end Behavioral;

**Test Bench Code:**

****

A <= '0'; B<= '0';

wait for 100 ns;

A <= '0'; B<= '1';

wait for 100 ns;

A <= '1'; B<= '0';

wait for 100 ns;

A <= '1'; B<= '1';

wait for 100 ns;

**Exercise#3:** Design **gate level logic diagram** of full adder (shown in Figure 3) in **structural style**

of modeling.

**Design code:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity FullAdder\_201b299 is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

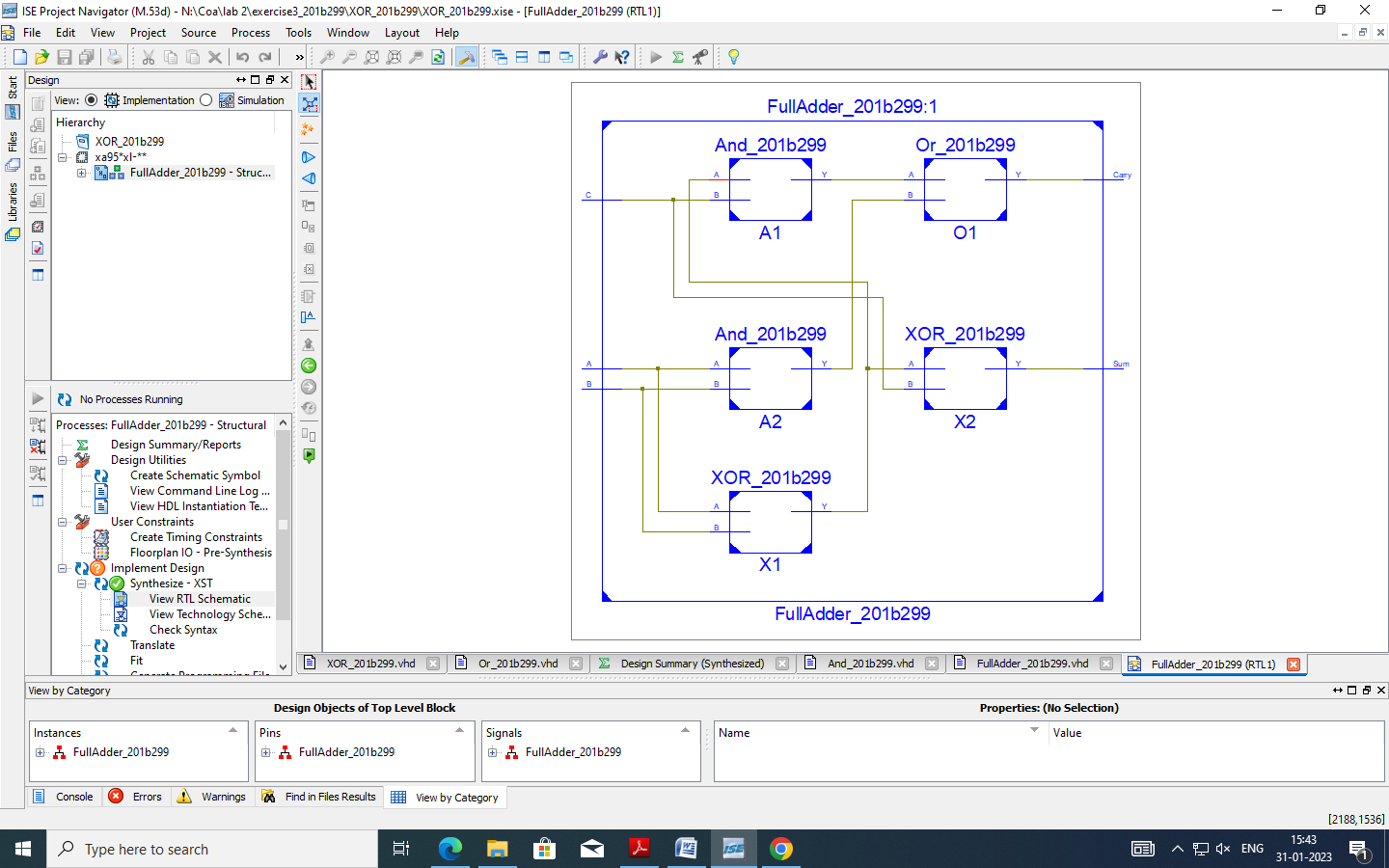
C : in STD\_LOGIC;

Sum : out STD\_LOGIC;

Carry : out STD\_LOGIC);

end FullAdder\_201b299;

architecture Structural of FullAdder\_201b299 is



component XOR\_201b299

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

Y : out STD\_LOGIC);

end component;

component Or\_201b299

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

Y : out STD\_LOGIC);

end component;

component And\_201b299

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

Y : out STD\_LOGIC);

end component;

signal s1, s2, s3 : std\_logic;

begin

X1 : XOR\_201b299 port map(A,B,s1);

X2 : XOR\_201b299 port map(s1,C,Sum);

A1 : And\_201b299 port map(s1,C,s2);

A2 : And\_201b299 port map (A,B,s3);

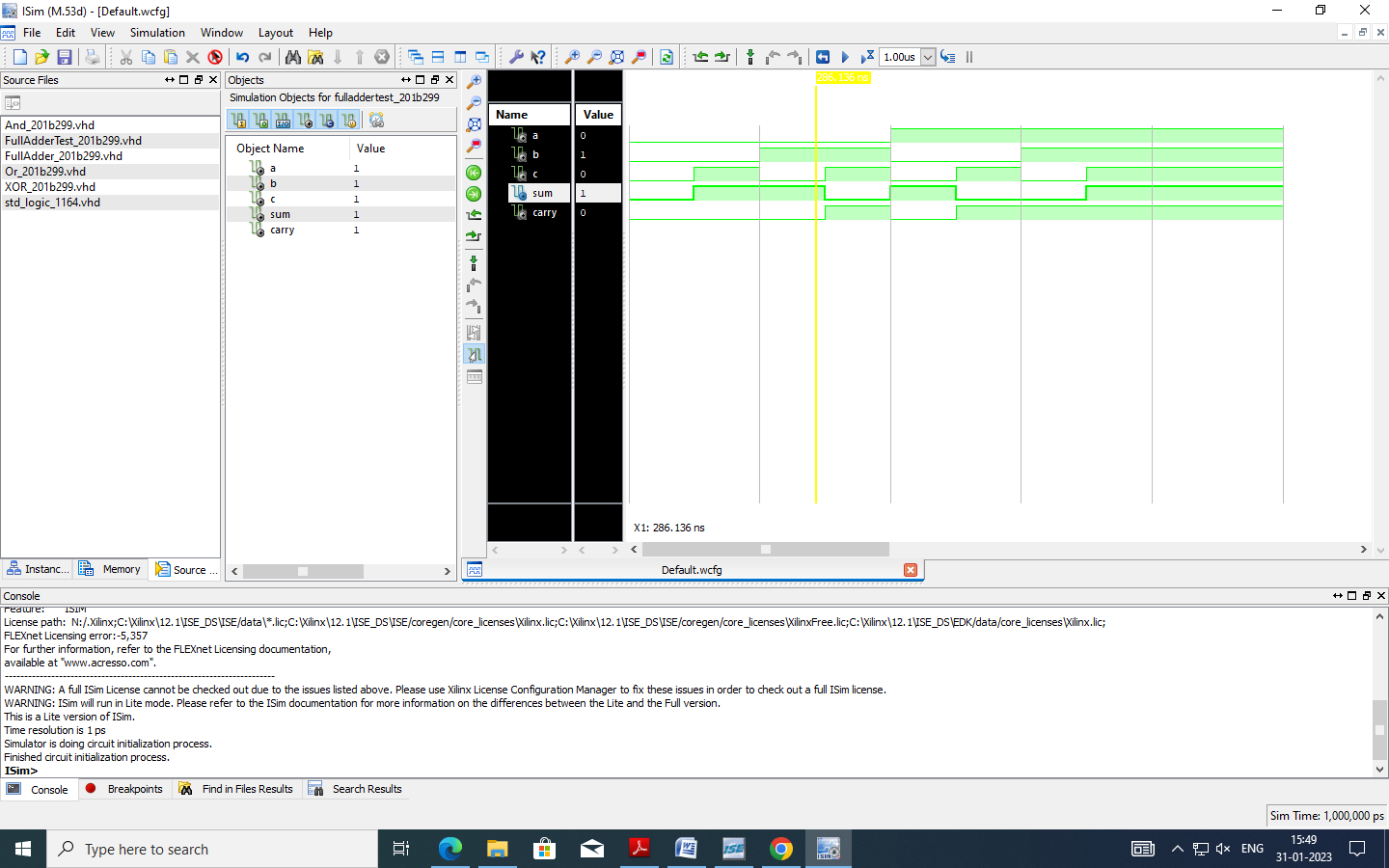
O1 : Or\_201b299 port map (s2,s3,carry);

end Structural;

**Test Bench Code:**

A <= '0'; B<= '0'; C<= '0';

wait for 100 ns;

A <= '0'; B<= '0'; C<= '1';

wait for 100 ns;

A <= '0'; B<= '1'; C<= '0';

wait for 100 ns;

A <= '0'; B<= '1'; C<= '1';

wait for 100 ns;

A <= '1'; B<= '0'; C<= '0';

wait for 100 ns;

A <= '1'; B<= '0'; C<= '1';

wait for 100 ns;

A <= '1'; B<= '1'; C<= '0';

wait for 100 ns;

A <= '1'; B<= '1'; C<= '1';

wait for 100 ns;