Course Name:	**Hardware Description Language Lab (2UXL401)**	**Semester:**	**IV**
Date of Performance:	**12/01/2021**	**Batch No:**	**B2**
Faculty Name:	**Prof. Bhargavi Kaslikar**	**Roll No:**	**1912060**
Faculty Sign & Date:		**Grade/Marks:**	

Experiment No: 1

Title: Study of basic VHDL code: Adder (Dataflow)

Aim and Objective of the Experiment:

Write a VHDL code

a) To implement a half adder.
b) A full adder using half adder
c) A 4-bit adder using full adder.

Write a testbench to verify your results for half adder and four bit adder Implement the full adder on CPLD.

To study basic structure of VHDL code and to understand use of test bench for simulation.
To know the process for implementation on CPLD.

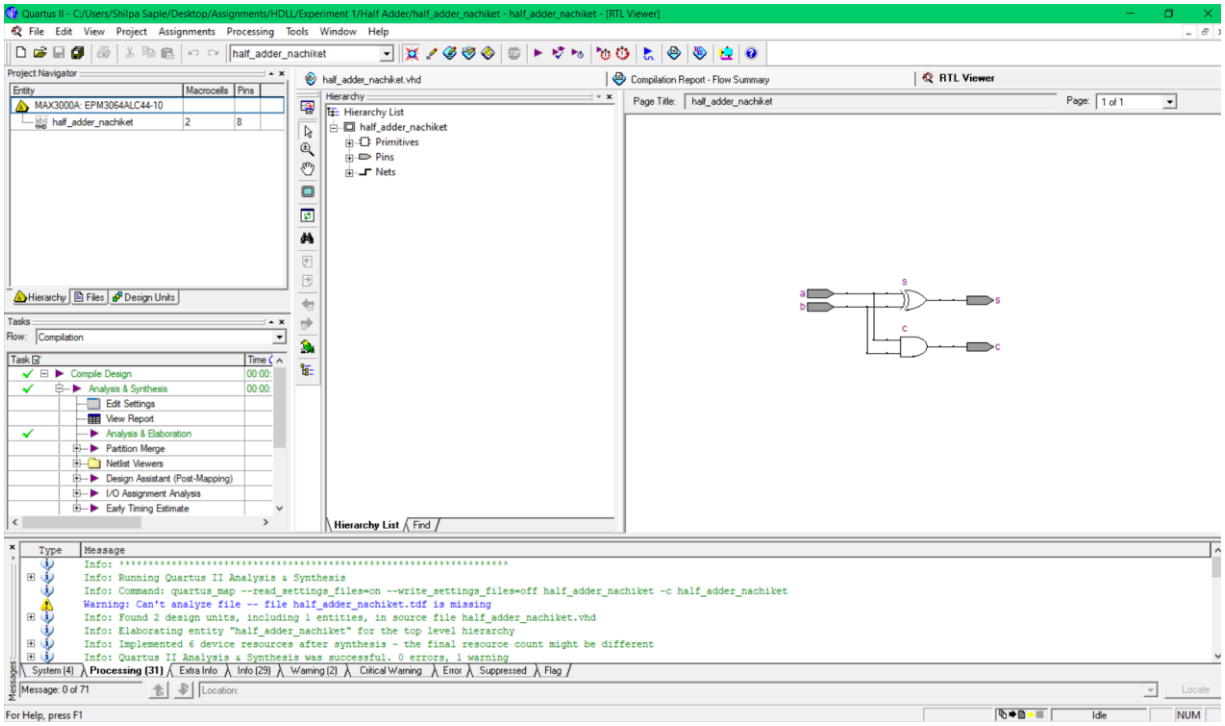
COs to be achieved:

CO 1: Use basic Concurrent and Sequential statements in VHDL and write codes for simple applications
CO 2: Test a VHDL code and verify the circuit model.
CO 3: Synthesize and Implement the designed circuits on CPLD/ FPGA.

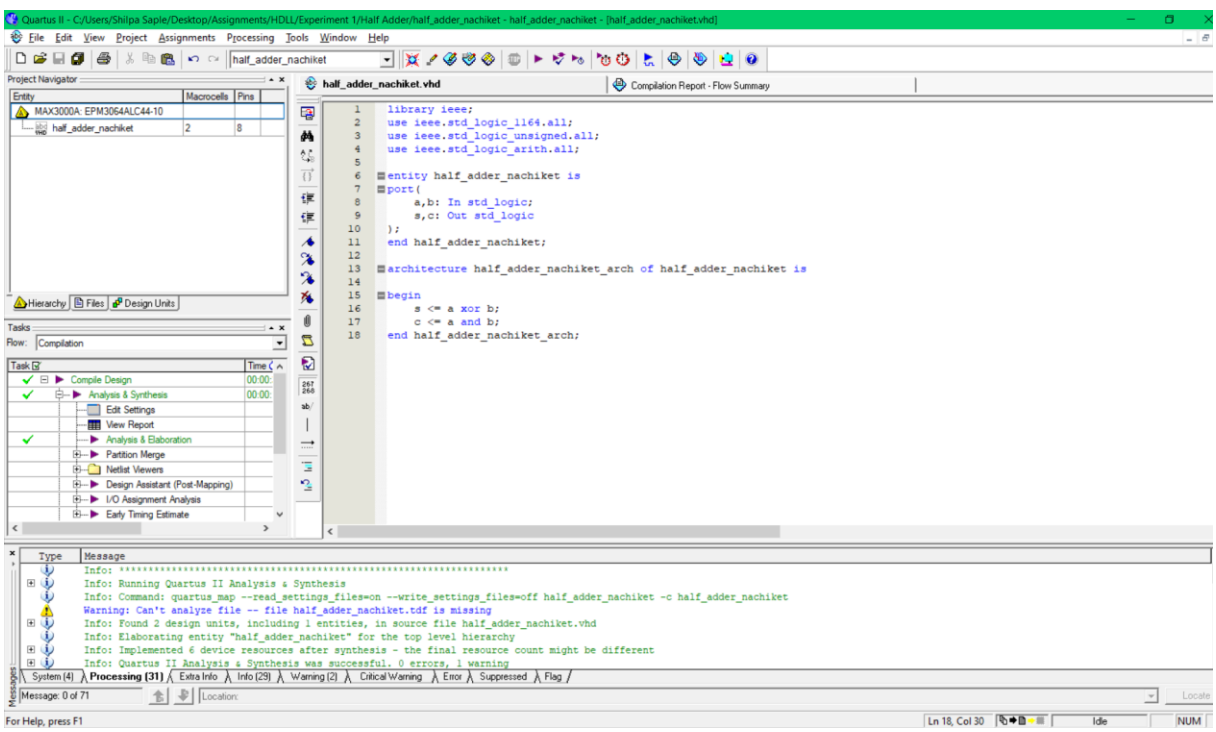
Work to be done

Upload VHDL codes for half adder, full adder (structural) and four-bit adder (structural) and test bench for half adder and 4-bit adder.
Also Upload Simulation waveforms for 4 bit adder.
Upload scanned image for post lab questions

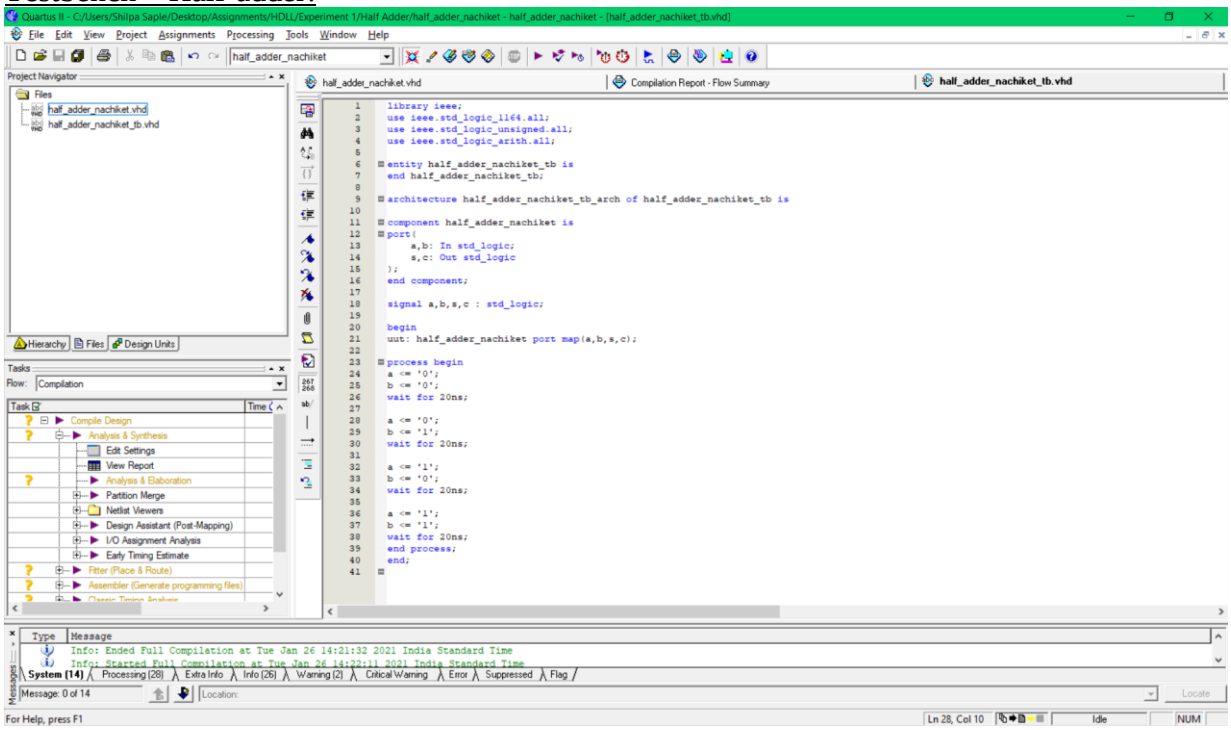
Half Adder:



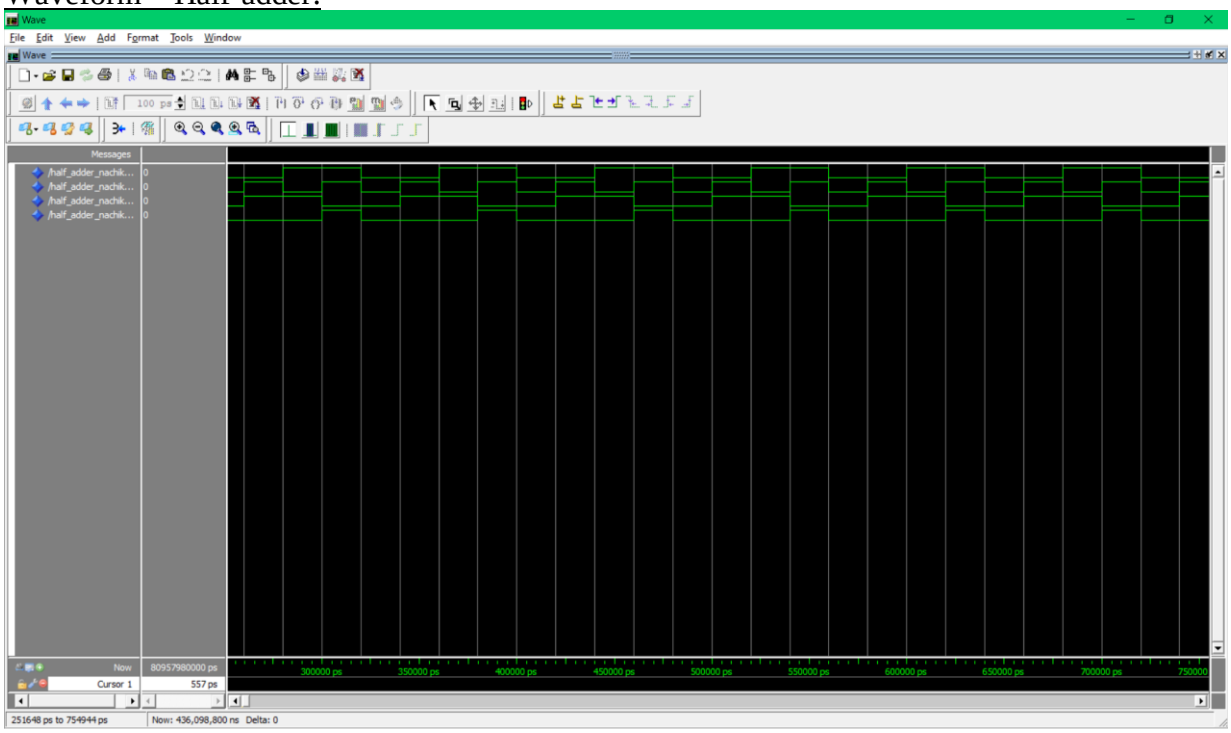
Entity – half-adder:



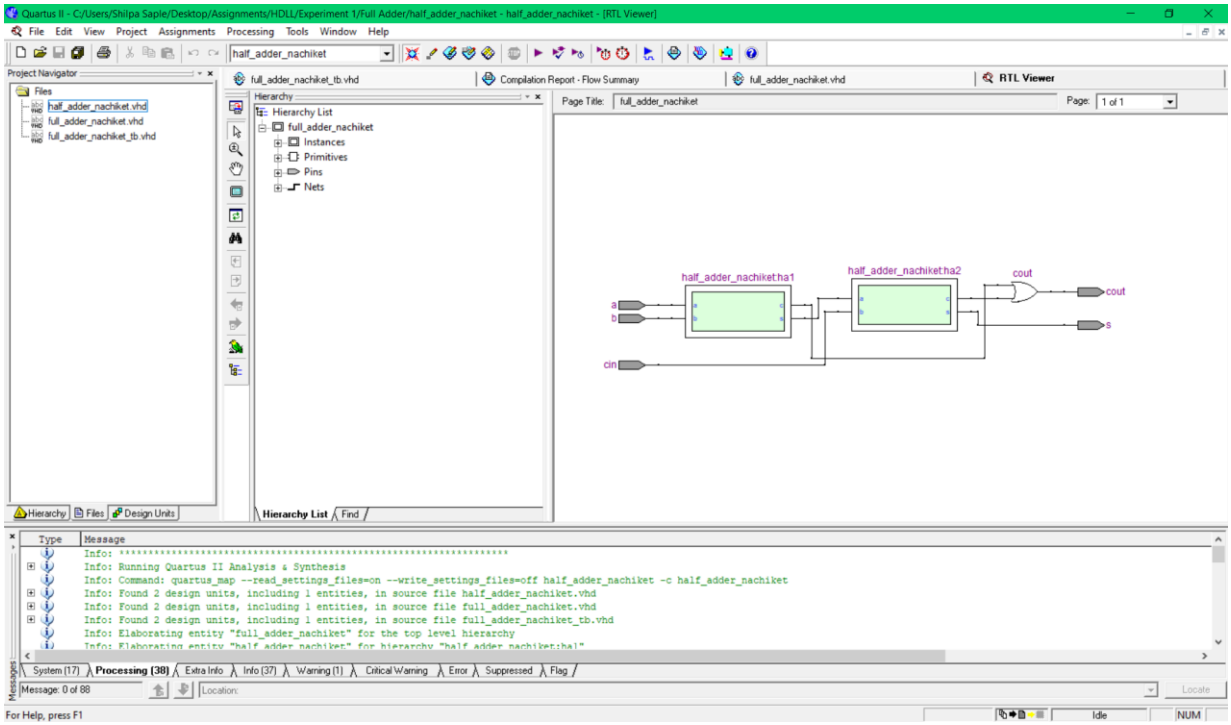
Testbench – Half-adder:



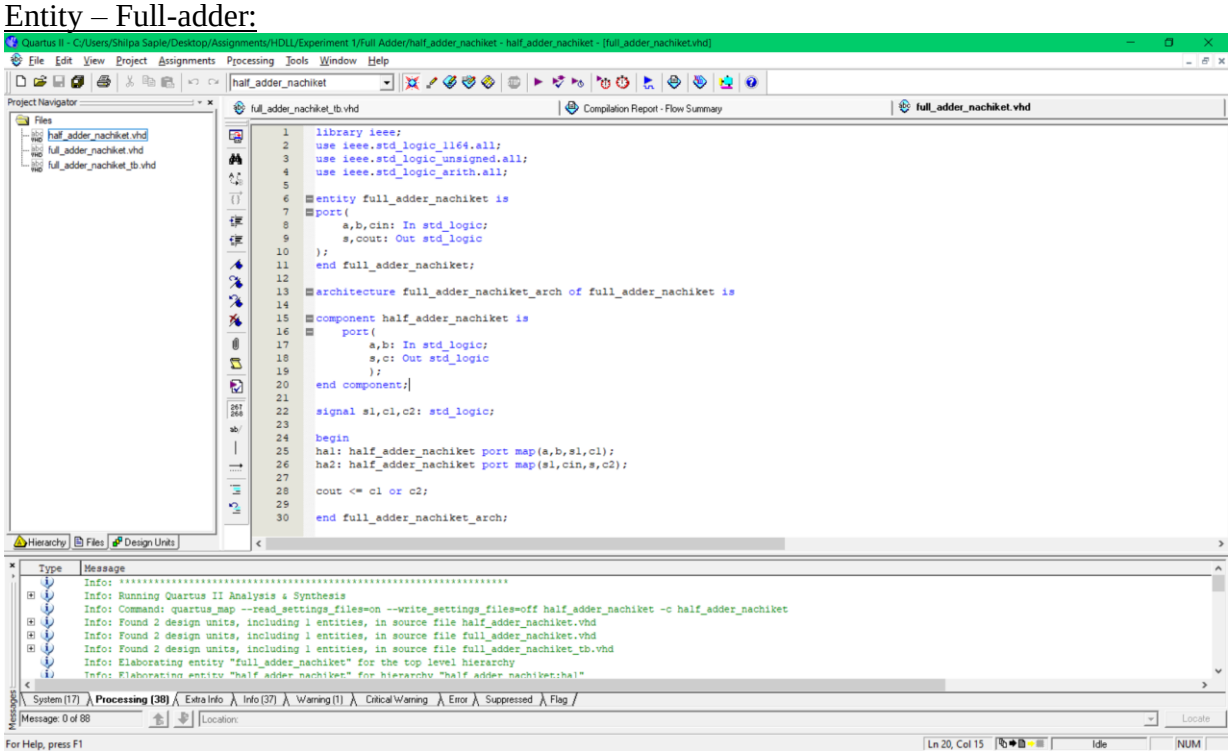
Waveform – Half-adder:



Full Adder:

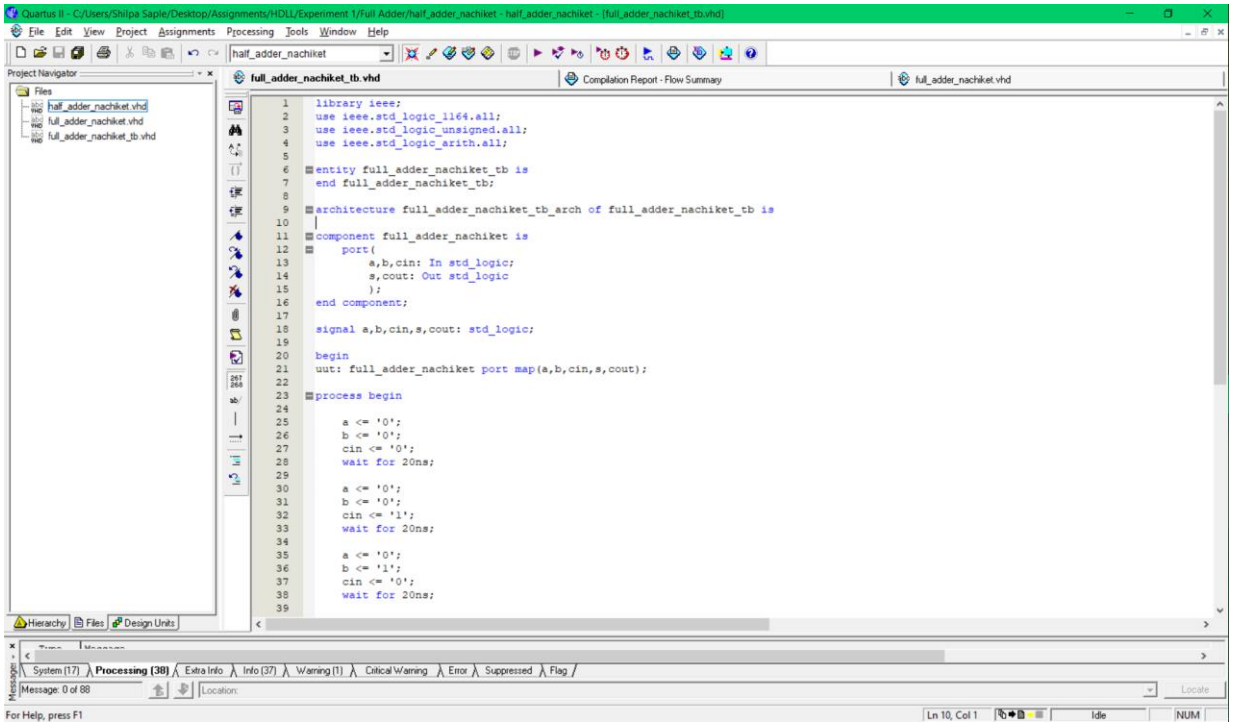


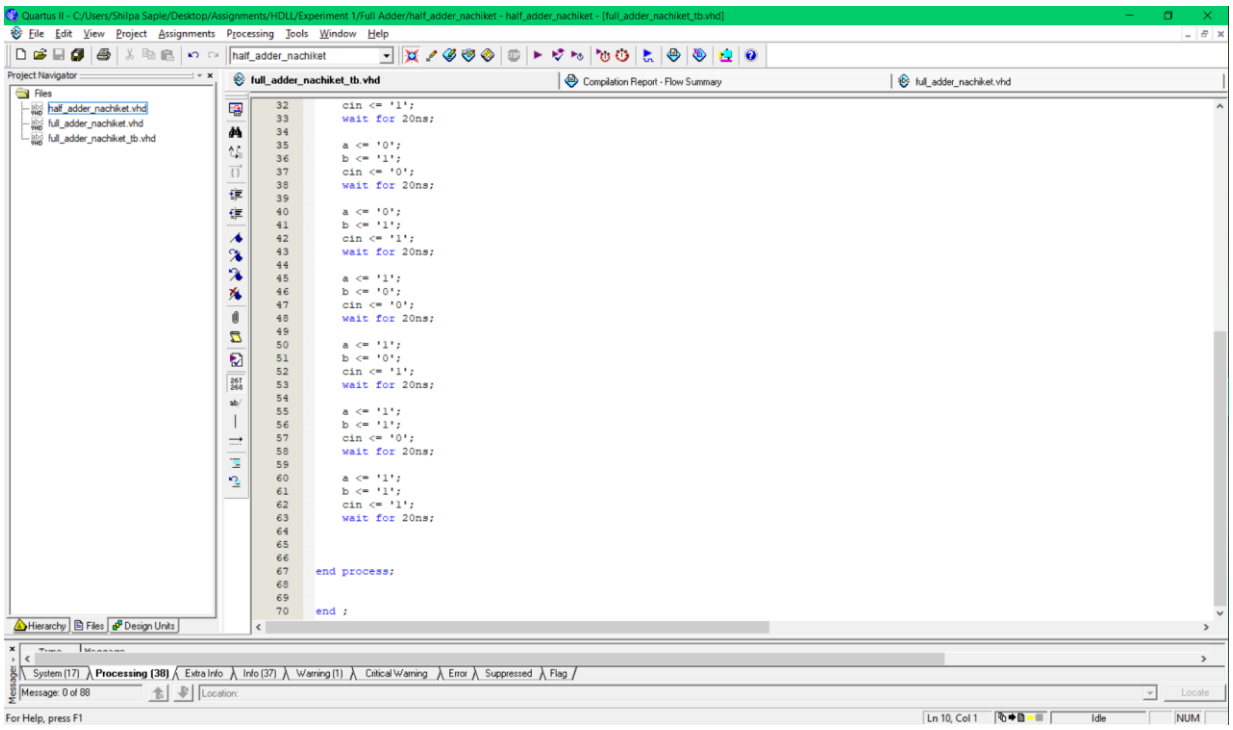
Entity – Full-adder:



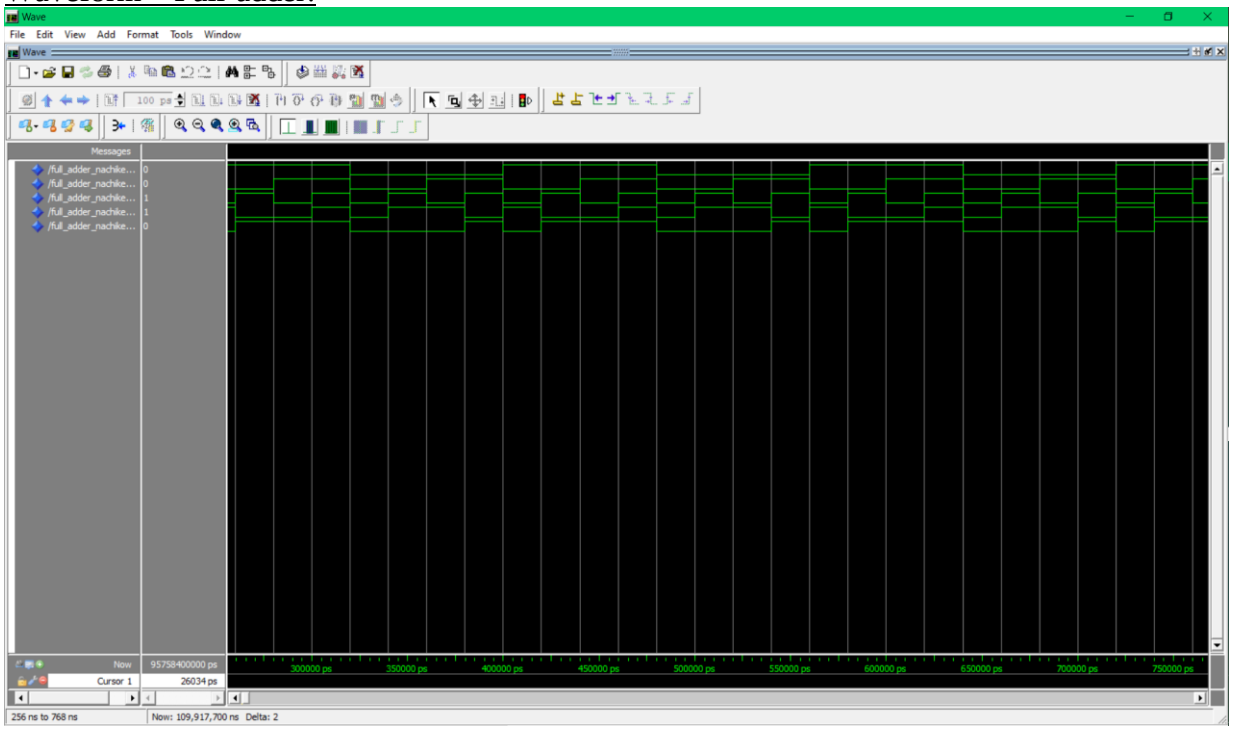
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity full_adder_nachiket is
port(
    a,b,cin: In std_logic;
    s,cout: Out std_logic
);
end full_adder_nachiket;

architecture full_adder_nachiket_arch of full_adder_nachiket is

component half_adder_nachiket is
    port(
        a,b: In std_logic;
        s,c: Out std_logic
        );
end component;

signal s1,c1,c2: std_logic;

begin
ha1: half_adder_nachiket port map(a,b,s1,c1);
ha2: half_adder_nachiket port map(s1,cin,s,c2);

cout <= c1 or c2;

end full_adder_nachiket_arch;
```

Testbench – Full-adder:

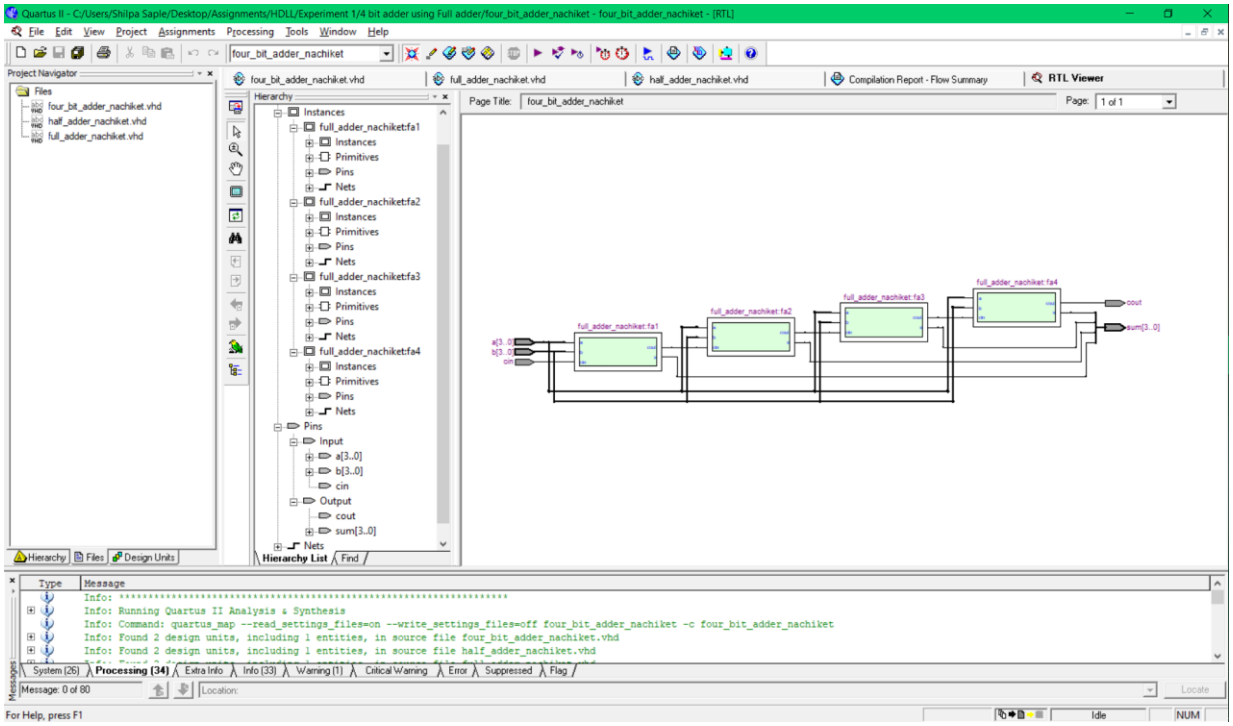




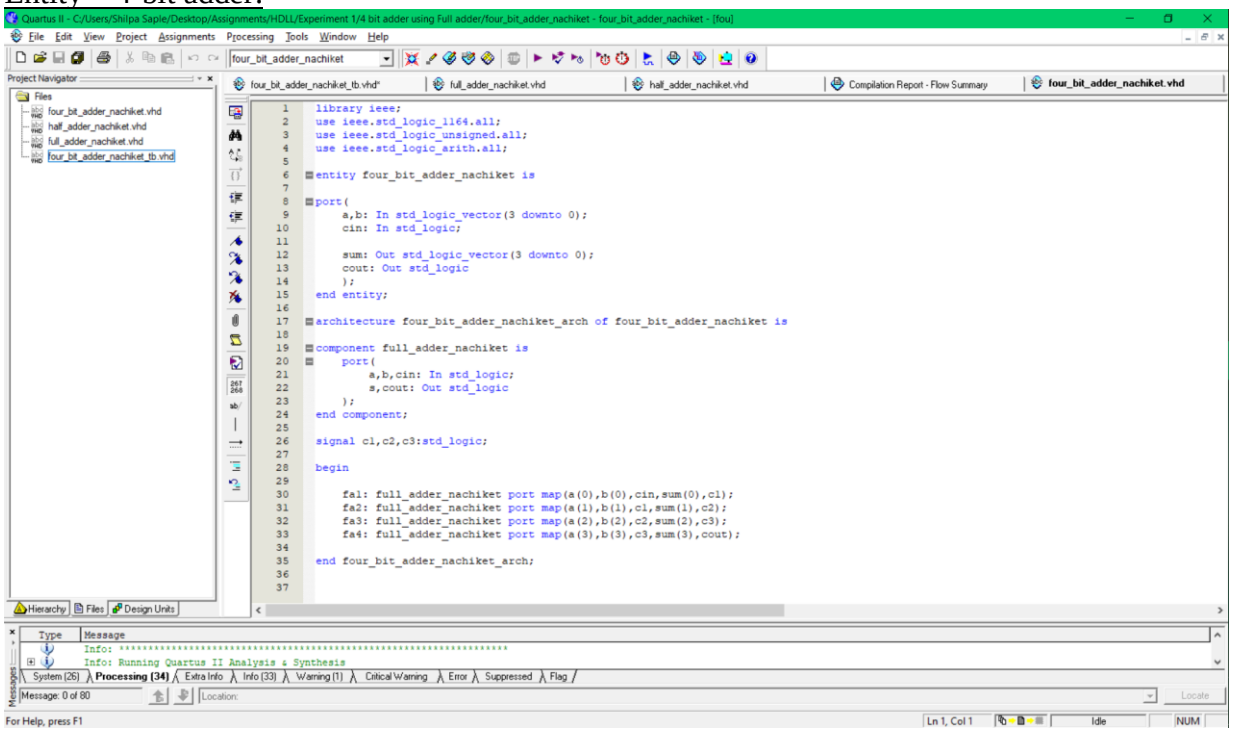
Waveform – Full-adder:



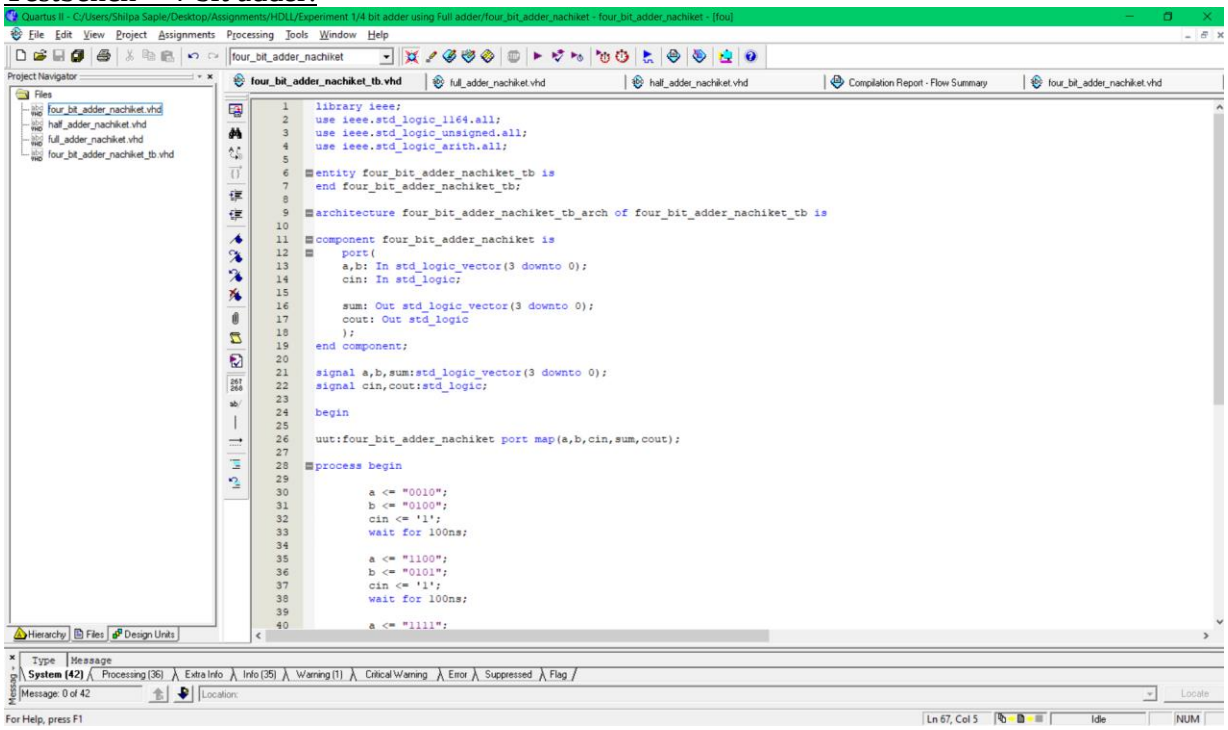
4-bit Adder using Full-adder:

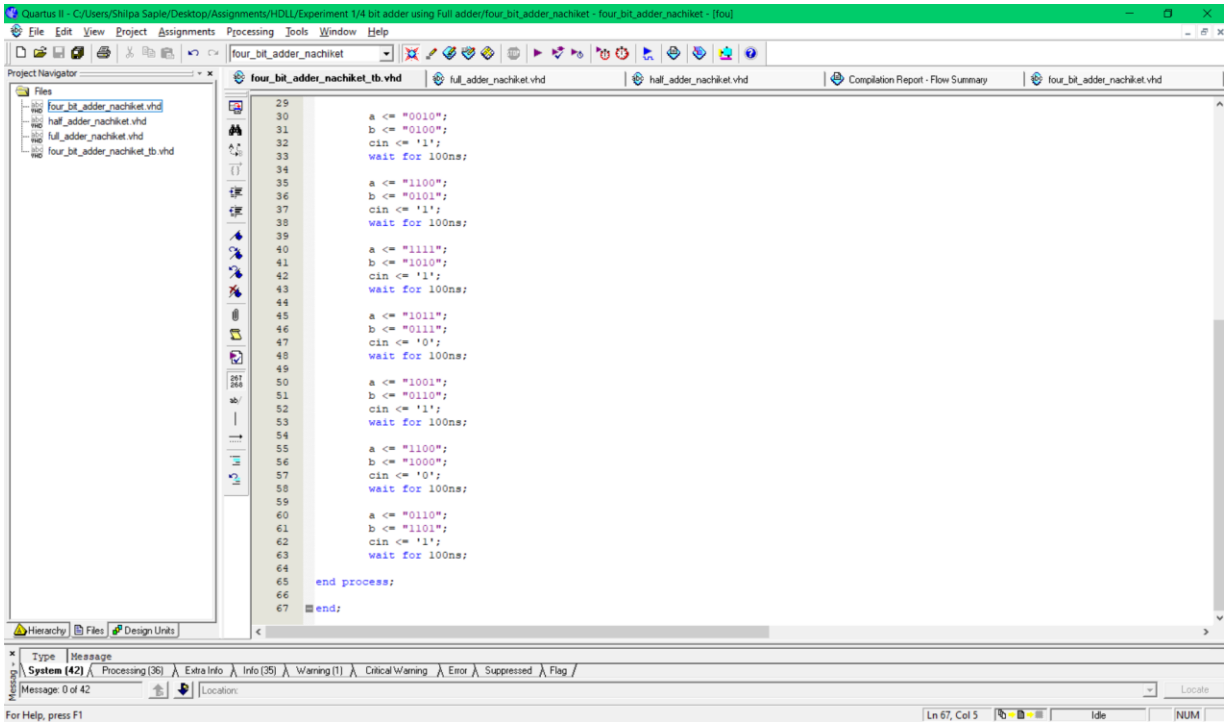


Entity – 4-bit adder:

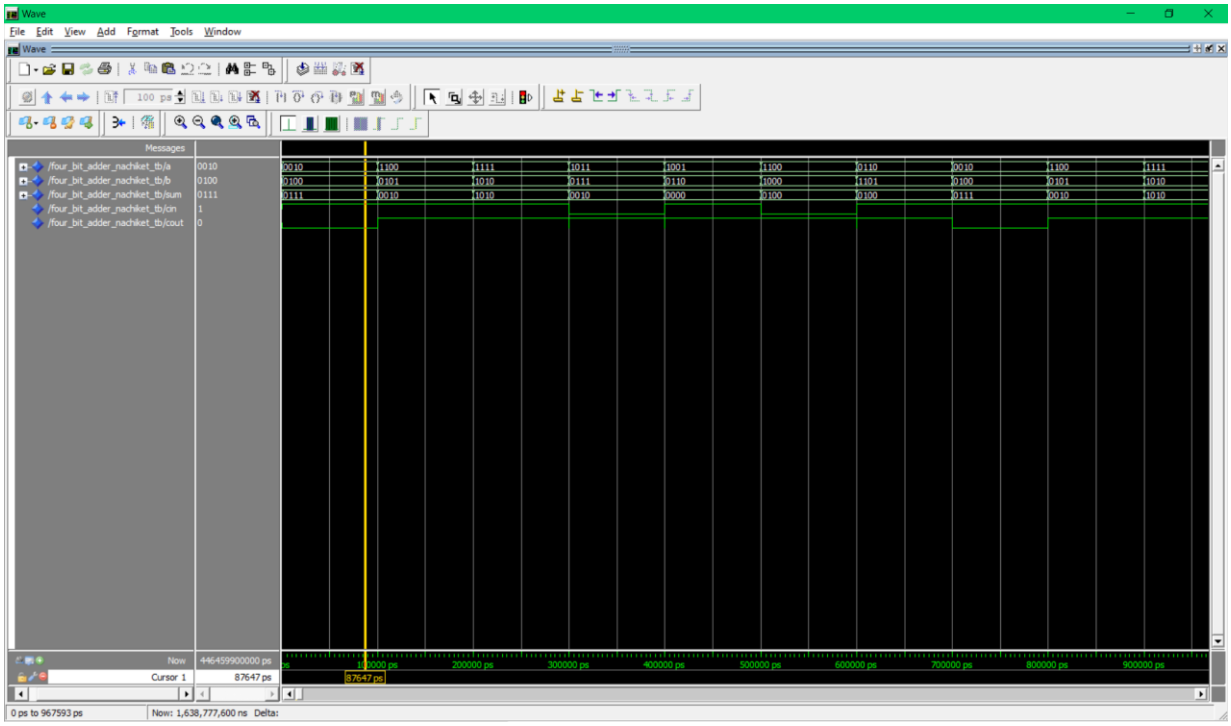


Testbench – 4-bit adder:





Waveform – 4-bit adder:



Post Lab Subjective/Objective type Questions:

Upload Answer of following question before coming to next laboratory.

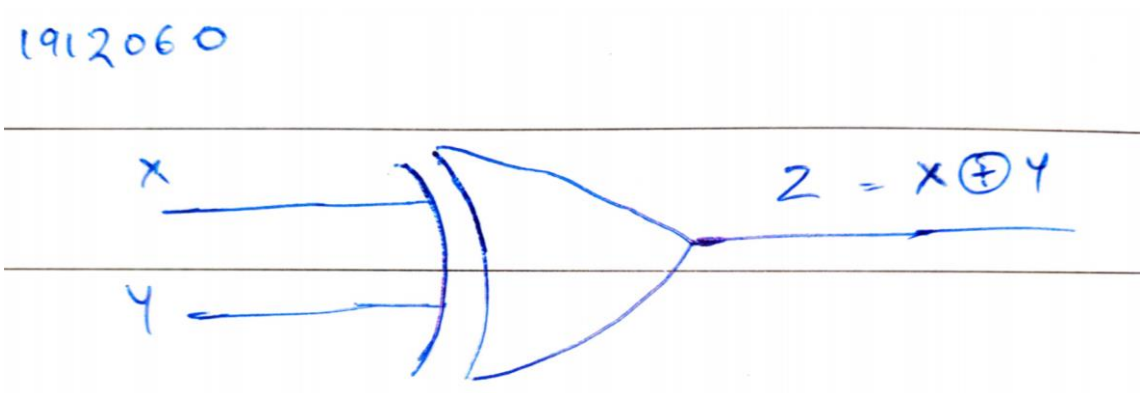
Q1. **Analyse the following code and write its output.**

```
library ieee;
use ieee.std_logic_1164.all;
entity xyz is
port ( x,y: in std_logic;
          z : out std_logic);
end entity;
architecture arch_xyz of xyz is
begin
z <= '0' when (x='1' and y='1') else
    '0'when (x='0' and y='0')else
    '1';
end arch_xyz;
```

Ans:

The given code is similar to XOR gate as the output ‘z’ is low (0) if both the inputs ‘x’ and ‘y’ are same i.e. if both of them are high or if both of them are low at the same time. Output is high if ‘x’ and ‘y’ are different.

x	y	**z**
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**



Q 2 Write a test bench for the above code

Ans:

1912060 – HDL Prac LAB Q 2

```
library ieee;
use ieee.std_logic_1164.all;
entity xyz_tb is
end xyz_tb;

architecture xyz_tb_arch of xyz_tb is
component xyz is
port(
        x,y : in std_logic;
        z : out std_logic);
end component;

signal x,y,z : std_logic;
begin
uut: xyz port map (x,y,z);
process begin
        x <= '0';
        y <= '0';
        wait for 20ns;

        x <= '0';
        y <= '1';
        wait for 20 ns;

        x <= '1';
        y <= '0';
        wait for 20 ns;

        x <= '1';
        y <= '1';

        wait for 20 ns;

end process;

end;
```

Conclusion:

In this experiment we learned the basics of VHDL and how to use Quartus and simulate the wave using Modelsim.
The VHDL code for Half Adder was written first, which was then used to make the Full adder. Then, the full adder was used in the coding of 4-bit adder.
We also learnt to write testbench for the circuit, in which we give various inputs to the circuit so that we can see the output for different conditions.

Signature of faculty in-charge with Date: