Course Name:	**Hardware Description Language Lab (2UXL401)**	**Semester:**	**IV**
Date of Performance:	**02/02/2021**	**Batch No:**	**B2**
Faculty Name:	**Prof. Bhargavi Kaslikar**	**Roll No:**	**1912060**
Faculty Sign & Date:		**Grade/Marks:**	

Experiment No: 2

Title: Study of Multiplexer (Dataflow, Behavioral, Structural)

Aim and Objective of the Experiment:

Write a VHDL code for implementing a 4:1 multiplexer

a) Using when else construct (Data Flow)

b) Using structural architecture

Write a testbench to verify your results.

Write a VHDL code for 16: 1 mux using 4:1 mux and gates. (Structural)

To study basic types of architectures and to understand the use of concurrent statements.

COs to be achieved:

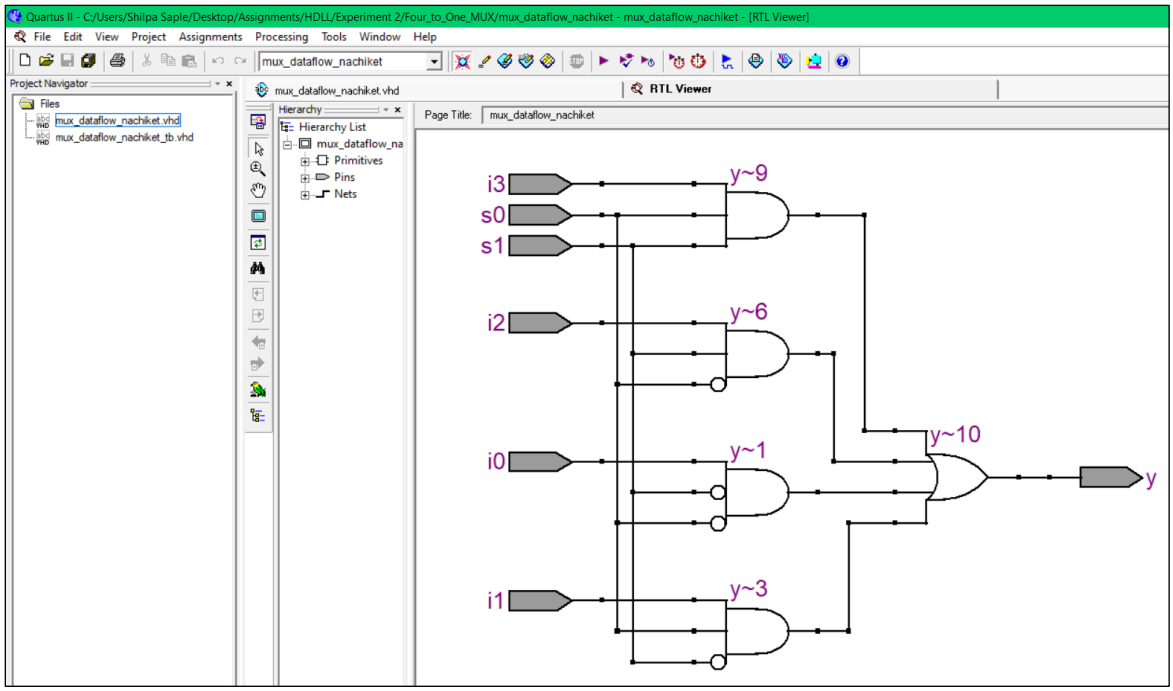
CO 1: Use basic Concurrent and Sequential statements in VHDL and write codes for simple applications

CO 2: Test a VHDL code and verify the circuit model.

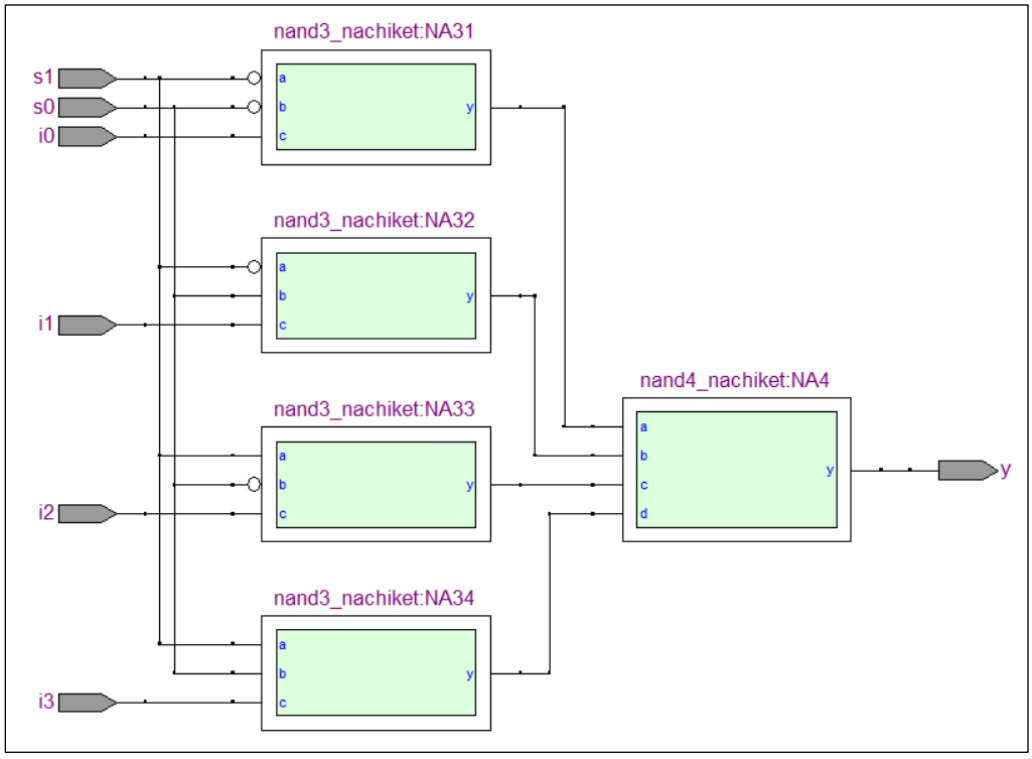
CO 3: Synthesize and Implement the designed circuits on CPLD/ FPGA.

Logic circuits:

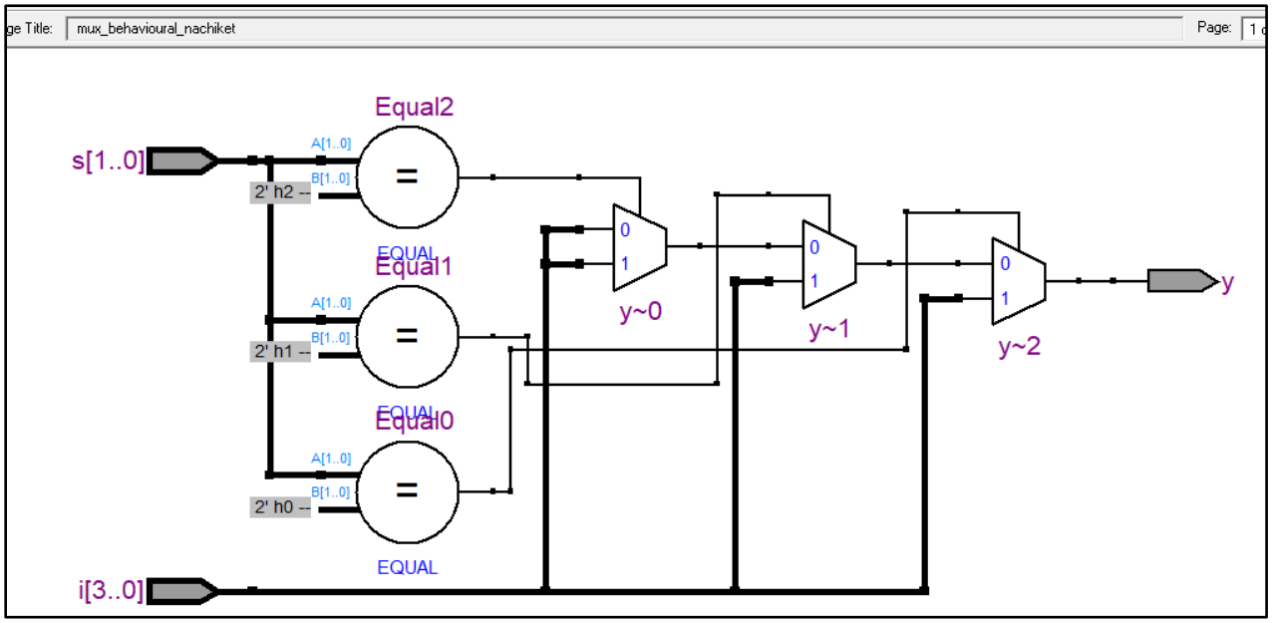
4:1 Mux – Dataflow



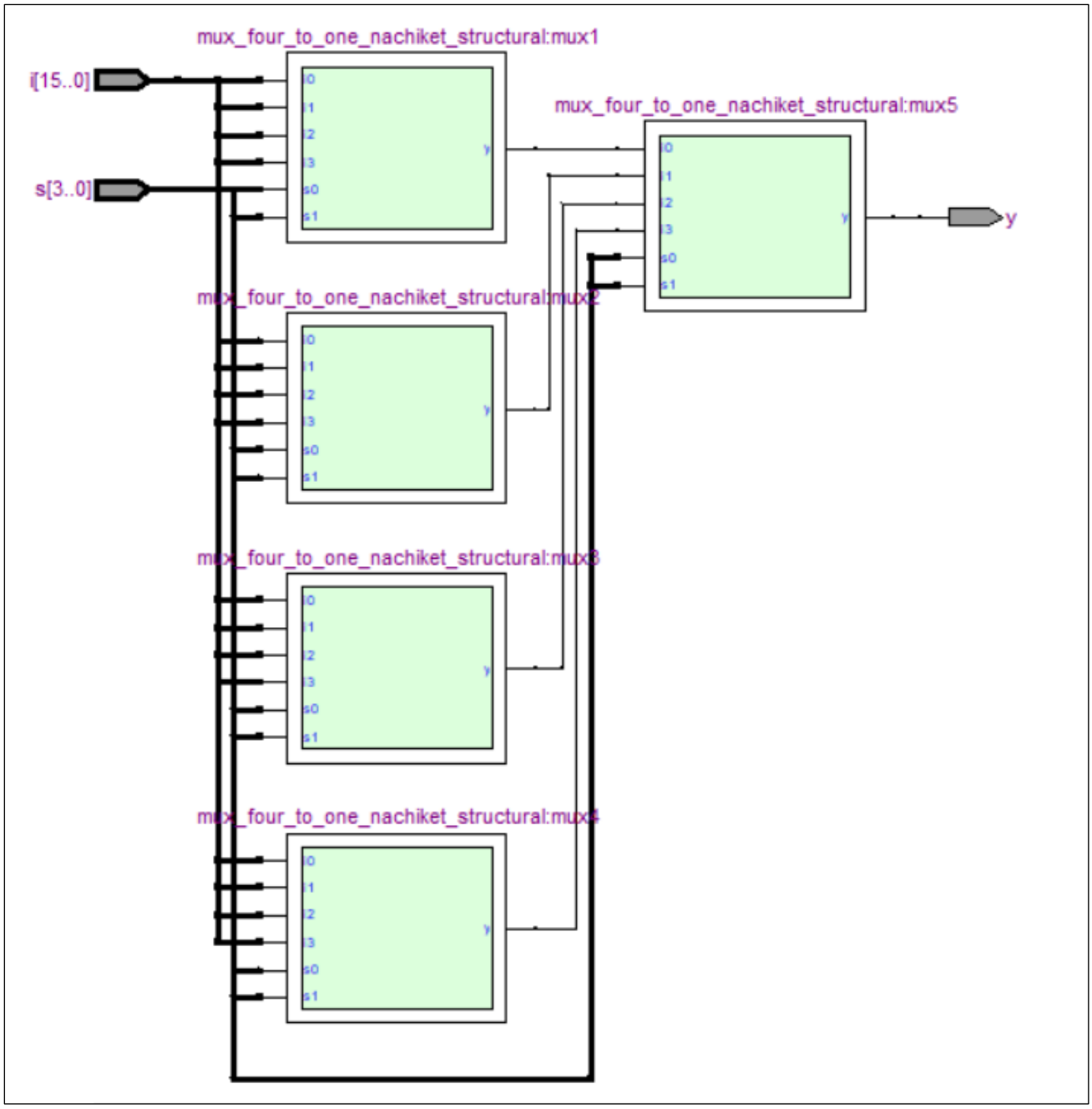
4:1 Mux – Structural



4:1 Mux – Behavioral



16:1 Mux – Structural



Work to be uploaded

Code for 4:1 Mux , Code for 16:1 mux

4:1 Data Flow:

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

entity mux_dataflow_nachiket is
port(
   i0,i1,i2,i3,s0,s1: In std_logic;
   y: Out std_logic
   );
   end mux_dataflow_nachiket;

architecture mux_dataflow_nachiket_arch of mux_dataflow_nachiket is
signal u1,u0:std_logic;
begin
   u1 <= not s1;
   u0 <= not s0;
   y<= (u1 and u0 and i0) or (u1 and s0 and i1) or (s1 and u0 and i2) or (s1 and s0 and i3) ;
 end mux_dataflow_nachiket_arch;
```

4:1 Structural:

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

entity mux_four_to_one_nachiket_structural is
        port(
                i0,i1,i2,i3,s0,s1: In std_logic;
                y: Out std_logic
                );
end mux_four_to_one_nachiket_structural;
architecture mux_four_to_one_nachiket_structural_arch of mux_four_to_one_nachiket_structural
is

        component nand3_nachiket is
                port(
                        a,b,c: In std_logic;
                        y: Out std_logic
                );
   end component;
   component nand4_nachiket is
                port(
                        a,b,c,d: In std_logic;
                        y: Out std_logic
                );
   end component;

signal u1,u0,o0,o1,o2,o3:std_logic;
begin
        u1 <= not s1;
   u0 <= not s0;

        NA31: nand3_nachiket port map(u1,u0,i0,o0);
        NA32: nand3_nachiket port map(u1,s0,i1,o1);
        NA33: nand3_nachiket port map(s1,u0,i2,o2);
        NA34: nand3_nachiket port map(s1,s0,i3,o3);
        NA4: nand4_nachiket port map(o0,o1,o2,o3,y);
end mux_four_to_one_nachiket_structural_arch;
```

4:1 Behavioral:

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

entity mux_behavioural_nachiket is
        port(
                i: In std_logic_vector(3 downto 0);
                s: In std_logic_vector(1 downto 0);
                y: Out std_logic
                );
        end mux_behavioural_nachiket;

architecture mux_behavioural_nachiket_arch of mux_behavioural_nachiket is
        begin
                process(s,i)
                        begin
                                if(s="00") then
                                        y <= i(0);
                                elsif(s="01") then
                                        y <= i(1);
                                elsif(s="10") then
                                        y <= i(2);
                                else
                                        y <= i(3);
                                end if;

                end process;
end mux_behavioural_nachiket_arch;
```

16:1 Mux using 4:1 Mux – Structural:

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

entity mux_16_to_one_nachiket is
	port(
		i: In std_logic_vector(15 downto 0);
		s: In std_logic_vector(3 downto 0);
		y: Out std_logic
		);
end mux_16_to_one_nachiket;

architecture mux_16_to_one_nachiket_arch of mux_16_to_one_nachiket is

	component mux_four_to_one_nachiket_structural is
			port(
				i0,i1,i2,i3,s0,s1: In std_logic;
				y: Out std_logic
			);
	end component;

signal o: std_logic_vector(3 downto 0);

begin

	mux1: mux_four_to_one_nachiket_structural port map(i(0),i(1),i(2),i(3),s(0),s(1),o(0));
	mux2: mux_four_to_one_nachiket_structural port map(i(4),i(5),i(6),i(7),s(0),s(1),o(1));
	mux3: mux_four_to_one_nachiket_structural port map(i(8),i(9),i(10),i(11),s(0),s(1),o(2));
	mux4: mux_four_to_one_nachiket_structural port map(i(12),i(13),i(14),i(15),s(0),s(1),o(3));

	mux5: mux_four_to_one_nachiket_structural port map(o(0),o(1),o(2),o(3),s(2),s(3),y);


end mux_16_to_one_nachiket_arch;
```

Test bench for 4:1 mux and 16:1 Mux and simulation waveform of the same.

4:1 Testbench:

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

entity mux_dataflow_nachiket_tb is
end mux_dataflow_nachiket_tb;

architecture mux_dataflow_nachiket_tb_arch of mux_dataflow_nachiket_tb is

component mux_dataflow_nachiket is
port(
    i0,i1,i2,i3,s0,s1: In std_logic;
    y: Out std_logic
    );
 end component;
 signal  i0,i1,i2,i3,s0,s1,y: std_logic;

 begin
     uut:  mux_dataflow_nachiket port map( i0,i1,i2,i3,s0,s1,y);

     process begin
        i0<='1';
        i1<='0';
        i2<='1';
        i3<='0';

        s0<='0';
        s1<='0';
        wait for 20ns;

        s0<='0';
        s1<='1';
        wait for 20ns;

        s0<='1';
        s1<='0';
```

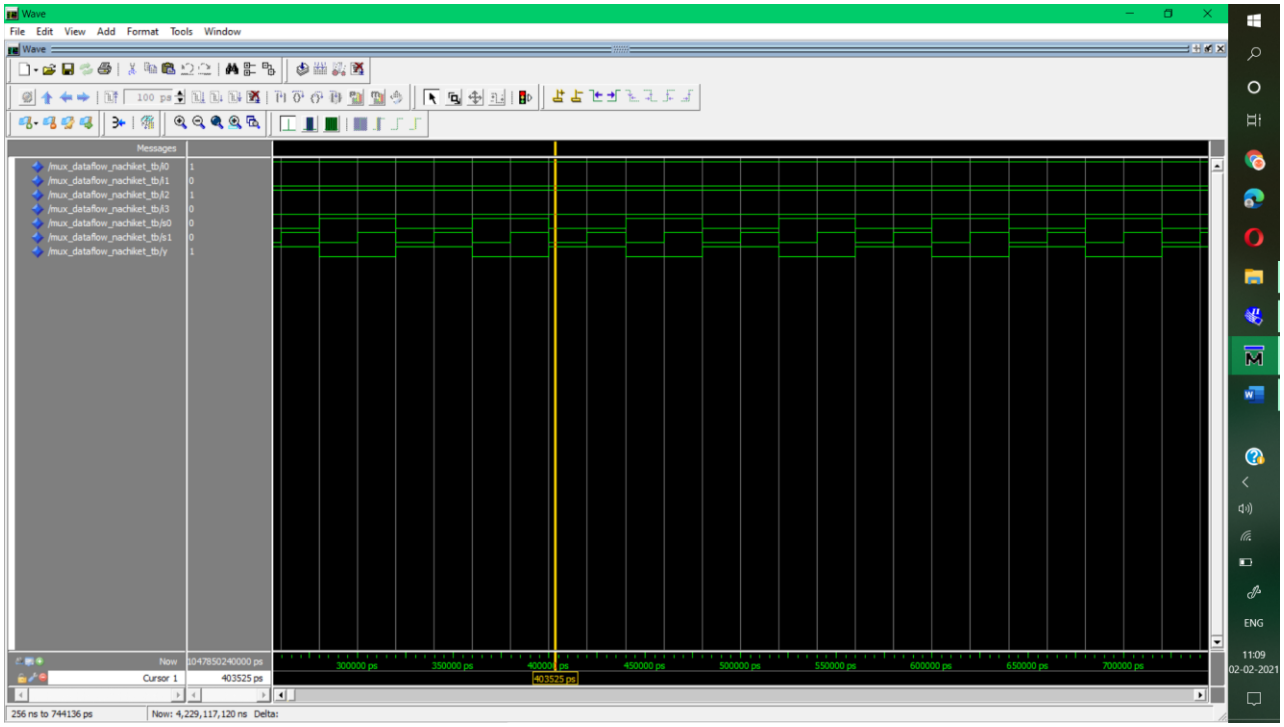
```
        wait for 20ns;

        s0<='1';
        s1<='1';
        wait for 20ns;
    end process;
 end;
```

4:1 Testbench Behavioral:

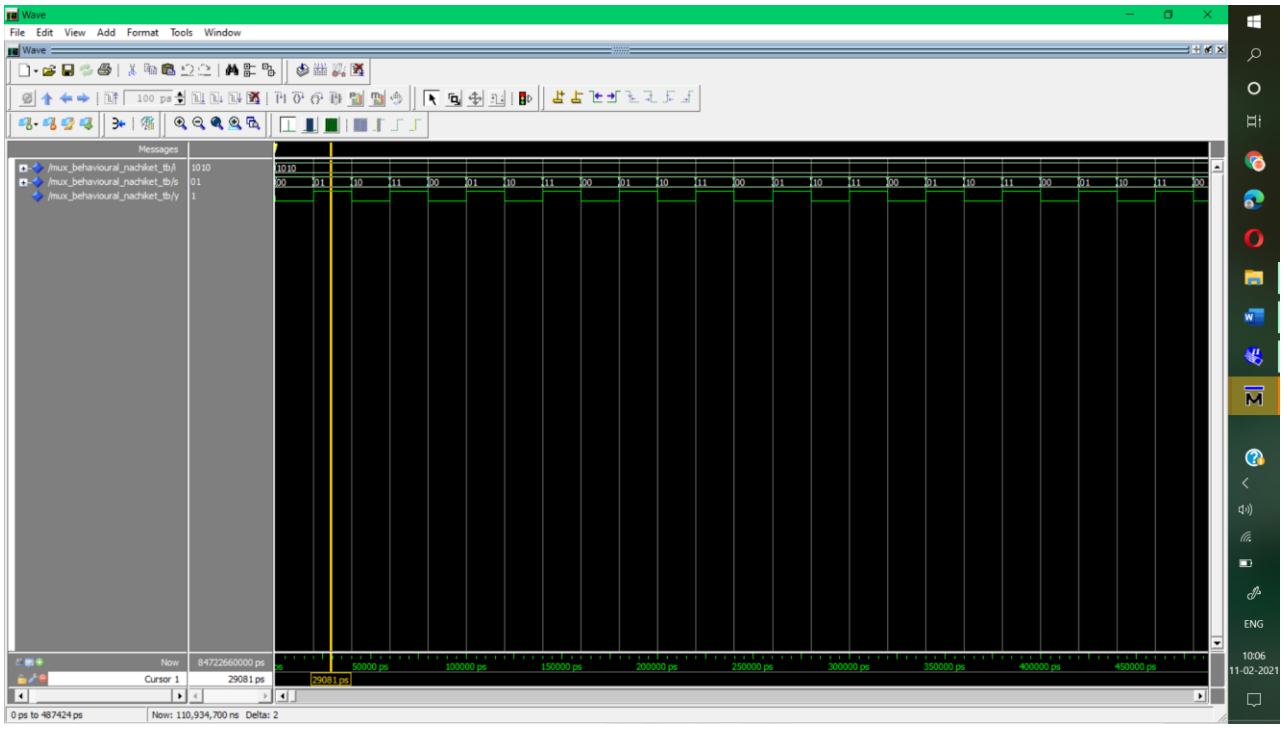
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity mux_behavioural_nachiket_tb is
end mux_behavioural_nachiket_tb;

architecture mux_behavioural_nachiket_tb_arch of mux_behavioural_nachiket_tb is

        component mux_behavioural_nachiket is
                port(
                        i: In std_logic_vector(3 downto 0);
                        s: In std_logic_vector(1 downto 0);
                        y: Out std_logic
                        );
        end component;

        signal i: std_logic_vector(3 downto 0);
        signal s: std_logic_vector(1 downto 0);
        signal y: std_logic;

        begin
                uut: mux_behavioural_nachiket port map(i,s,y);

                process begin
                        i <= "1010";

                        s <= "00";
                        wait for 20ns;
```

```
                s <= "01";
                wait for 20ns;

                s <= "10";
                wait for 20ns;

                s <= "11";
                wait for 20ns;

        end process;
end;
```

4:1 MUX waveform (dataflow):



4:1 Waveform (Behavioral):

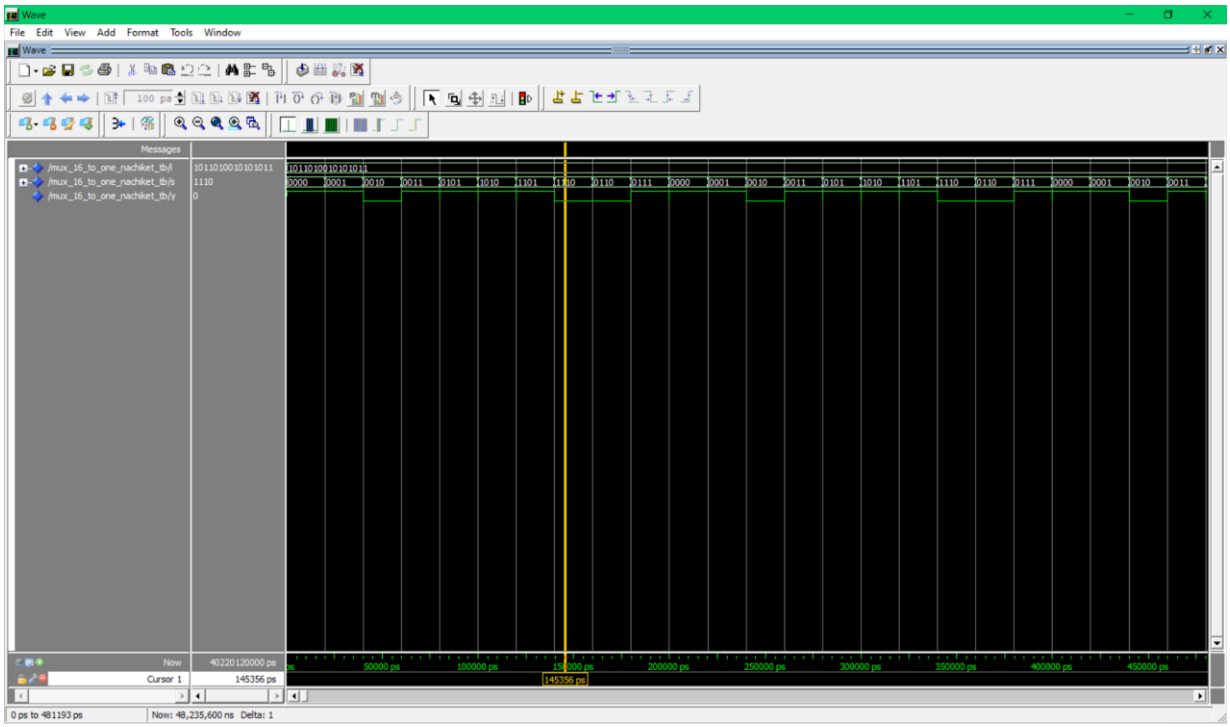


16:1 Testbench:

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

entity mux_16_to_one_nachiket_tb is
end mux_16_to_one_nachiket_tb;

architecture mux_16_to_one_nachiket_tb_arch of mux_16_to_one_nachiket_tb is

	component mux_16_to_one_nachiket is
		port(
			i: In std_logic_vector(15 downto 0);
			s: In std_logic_vector(3 downto 0);
			y: Out std_logic
			);
	end component;
	signal i: std_logic_vector(15 downto 0);
	signal s: std_logic_vector(3 downto 0);
	signal y: std_logic;

	begin
		uut:  mux_16_to_one_nachiket port map(i,s,y);

		process begin
			i <= "1011010010101011";

			s <= "0000";
			wait for 20ns;

			s <= "0001";
			wait for 20ns;
```

```
                s <= "0010";
                wait for 20ns;

                s <= "0011";
                wait for 20ns;

                s <= "0101";
                wait for 20ns;

                s <= "1010";
                wait for 20ns;

                s <= "1101";
                wait for 20ns;

                s <= "1110";
                wait for 20ns;

                s <= "0110";
                wait for 20ns;

                s <= "0111";
                wait for 20ns;


        end process;
end mux_16_to_one_nachiket_tb_arch;
```

16:1 Waveform:



Scanned copy of post lab questions

Post Lab Subjective/Objective type Questions:

Upload Answer of following question before coming to next laboratory.

Q1. **Analyze the following code and explain its output.**

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

entity xyz is
                port(g1,g2,g3:in std_logic;
                       sel: in std_logic_vector(2 downto 0);
                       q:out std_logic_vector(7 downto 0)
                       );
end xyz;

architecture xyz_a of xyz is
           signal q_s:  std_logic_vector(7 downto 0);
Begin
    with sel select q_s<=
           "01111111" when "000",
           "10111111" when "001",
           "11011111" when "010",
           "11101111" when "011",
           "11110111" when "100",
           "11111011" when "101",
           "11111101" when "110",
           "11111110" when "111",
           "11111111" when others;
     q<=q_s when (g1 and not g2 and not g3)= '1'

     else "11111111";

end xyz_a_a;
```

Page No.:
Date:

NACHIKET NAIK - 1912060 - B2

HDL - EXP-2 - POSTLAB

Q1)

~~Mistake~~ Error in the code :-

↳ Last line is given as end xyz-a-a; but it should be ~~xyz~~ end xyz-a;

- The given entity xyz has 4 inputs, g1, g2, g3 (std-logic i.e '0' or '1') and sel (vector of 3 bits) and one output q of 8 bits.

- In the code we are also using a signal q-s which is a vector of 8 bits

- The value of q-s depends on that of sel.

- Whatever be the value of sel, that bit of q-s which is at the corresponding decimal value ~~index~~ of sel will be 0. Rest 7 bits will be 1.

Indexing starts from 0.

eg: if sel = "101" → $(5)_{10}$ then the 6th bit of q-s will be 0 & others will be 1 ∴ q-s = "11111011"

This happens for sel in the range "000" to "111"

For any other value of sel, q-s = "11111111"

1912060

- Now, if the expression (g1 and not g2 and not g3) = '1' then the output q = q-s else q will get the value "11111111"
- The expression is true only when g1 = 1, g2 = 0, g3 = 0.

Q.2 Write a test bench for the above code.

1912060

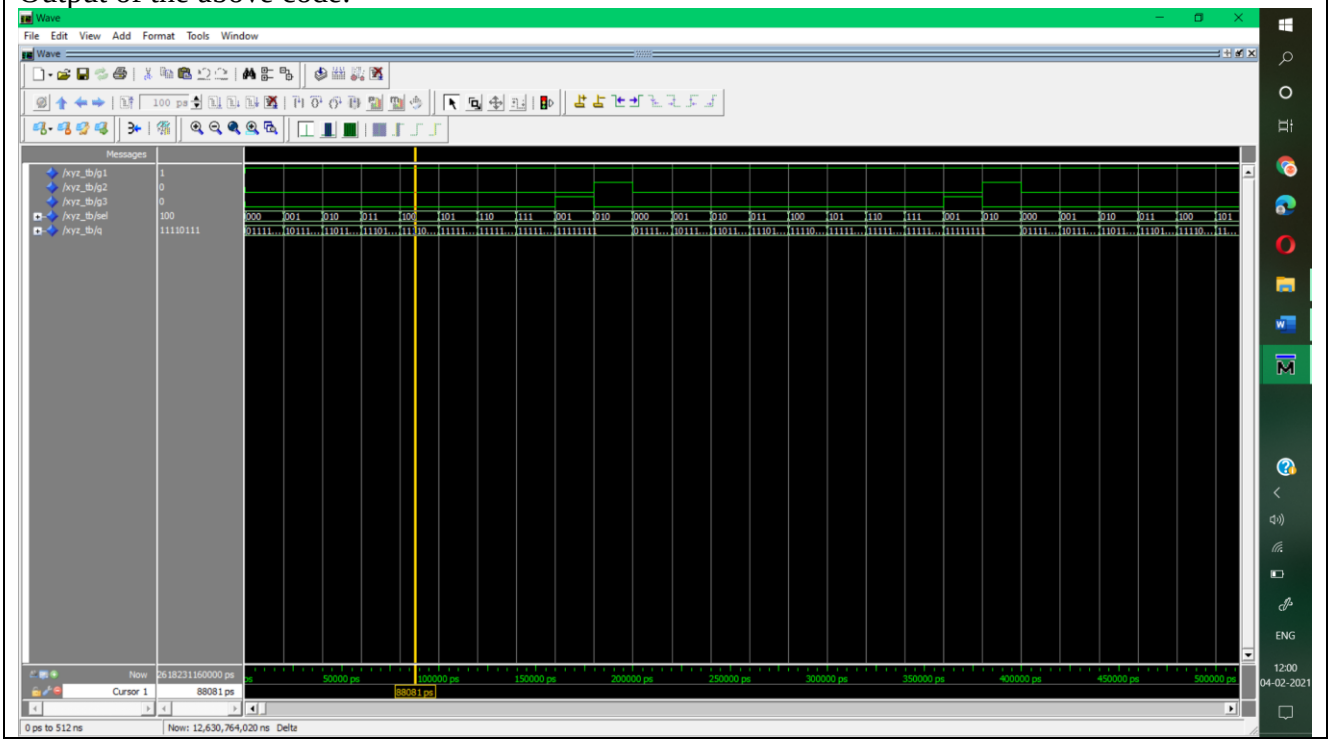
Q.2) Testbench :-

```
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 (
        g1, g2, g3: in std_logic;
        sel: in std_logic_vector (2 downto 0);
        q: out std_logic_vector (7 downto 0)
    );
end component;
signal g1, g2, g3: std_logic;
signal sel: std_logic_vector (2 downto 0);
signal q: std_logic_vector (7 downto 0);

begin
    uut: xyz port map (g1, g2, g3, sel, q);
    process begin
```

```
g1 <= '1';
g2 <= '0';
g3 <= '0';

sel <= "000";
wait for 20 ns;

sel <= "001"
wait for 20 ns;

sel <= "010";
wait for 20 ns;

sel <= "011";
wait for 20 ns;

sel <= "100";
wait for 20ns;

sel <= "101";
wait for 20 ns;

sel <= "110";
wait for 20 ns;

sel <= "111"
wait for 20 ns;

g3 <= '1';
sel <= "001";
wait for 20 ns;
```

```
g3 <= '0';
g2 <= '1';
sel <= "010";
wait for 20 ns;
end process
end xyz_tb_arch;
```

Output of the above code:



Conclusion:

NACHIKET NAIK - 1912060 - B2
HDL - EXP 2
CONCLUSION

↳ A multiplexer or MUX is a circuit that selects a single output from multiple inputs.

↳ In this experiment we wrote the code for 4:1 ~~mi~~ by using both data flow and structural architecture.

↳ In dataflow, we specify the functionality of an entity by defining the flow of information.

↳ In structural architecture, the various components needed are declared first, then instances of components created with particular mappings of signal corresponds ~~the~~ to the various pins of components.

↳ Hence in dataflow of 4:1 mux, we used the SOP equation where as in structural, we created 2 more components : 3 input & 4 input NAND gates.

↳ Lastly, we implemented a 16:1 mux using 5, 4:1 mux (mux tree). Hence we used structural architecture as the circuit made use of predefined components.

Signature of faculty in-charge with Date: