KURUKULASOORIYA KVR **200332X**

COMBINATIONAL CIRCUITS

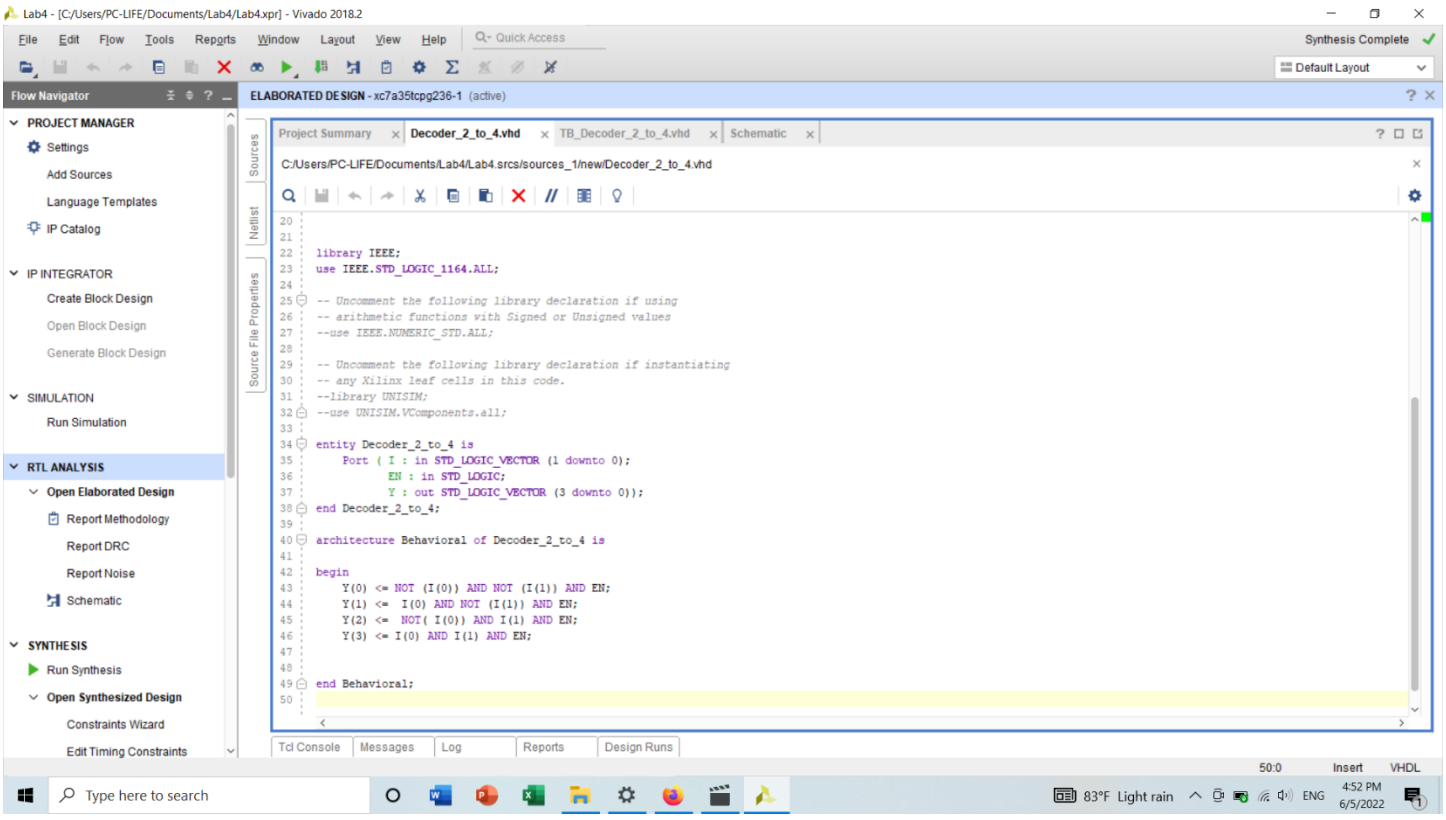
LAB 04 REPORT

Task Assigned:

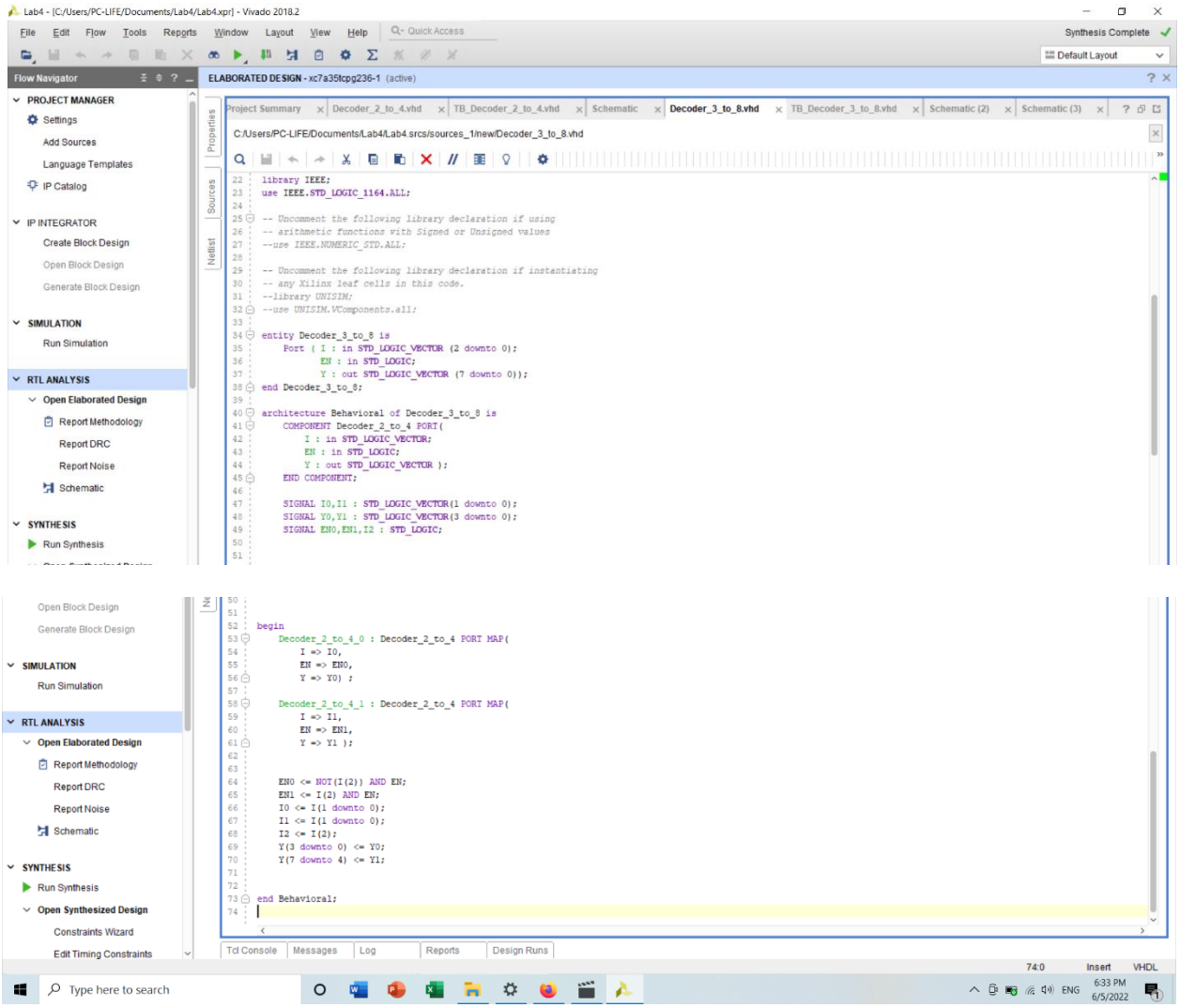
1. Analyze the functionality of decoder and multiplexer.
2. Design and develop a 2 to 4 decoder in VHDL using Vivado software.
3. Design and develop 3 to 8 decoder using 2 of above 2 to 4 decoders.
4. Design and develop 8 to 1 multiplexer using the 3 to 8 decoder.
5. Verify the functionality of each of above 3 circuits via a simulation using input combinations as per the numbers in the index number

VHDL Source Codes

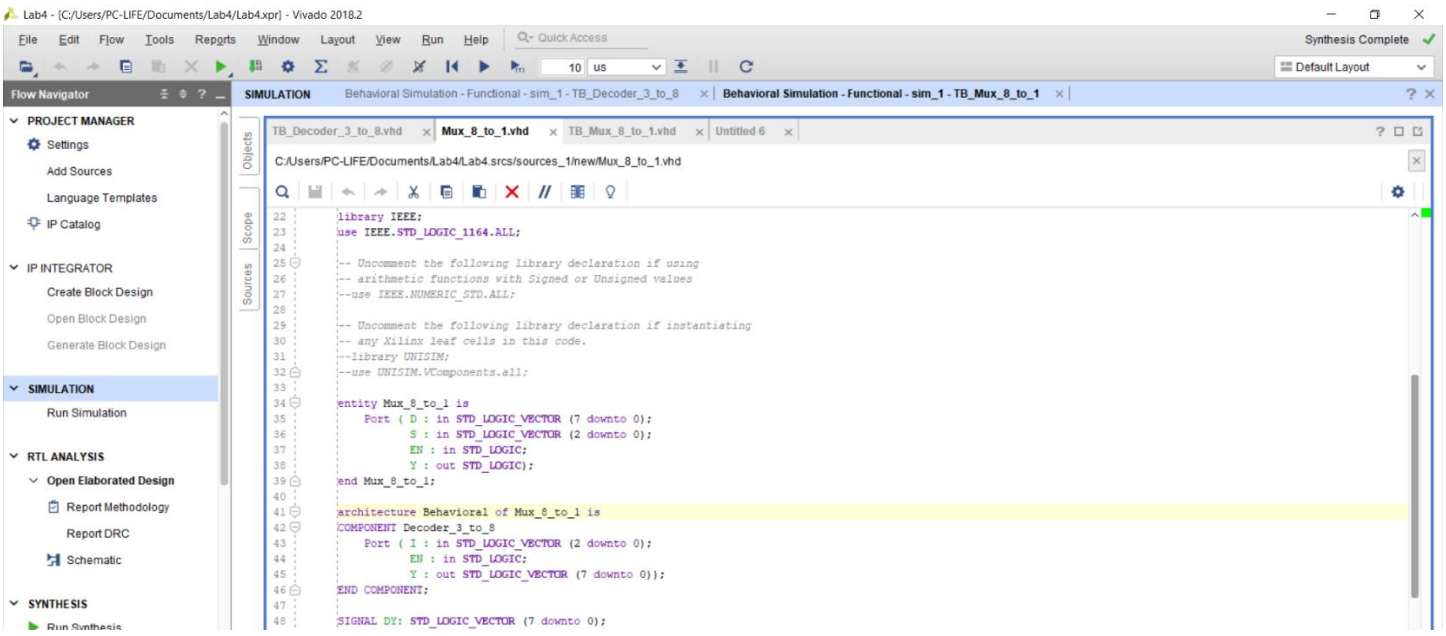
01.VHDL Code for 2 to 4 decoder

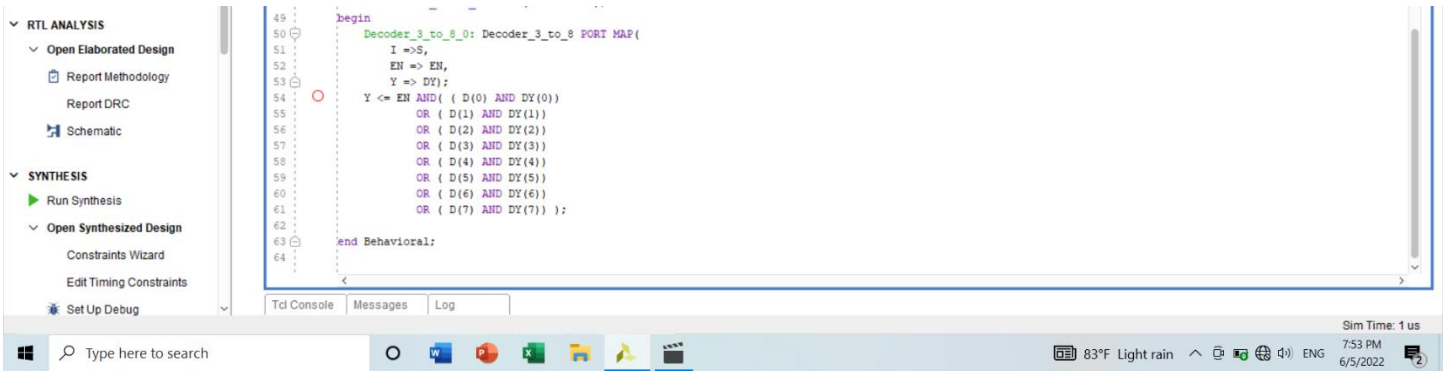


02.VHDL Code for 3 to 8 decoder



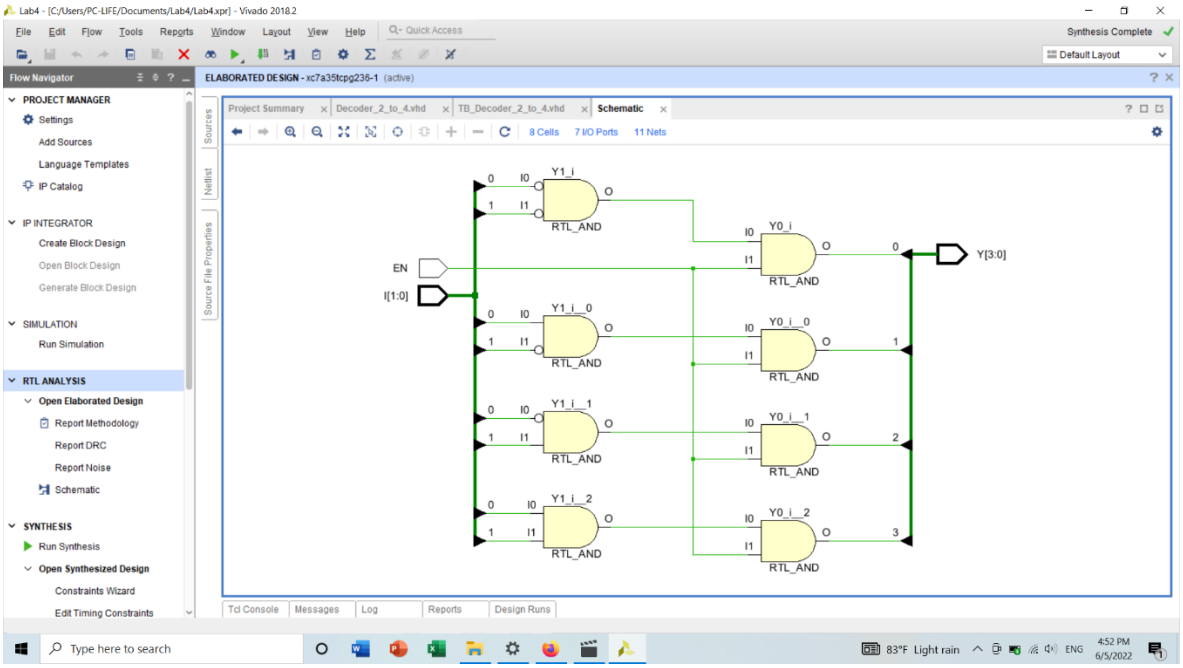
03.VHDL Code for 8 to 1 multiplexer



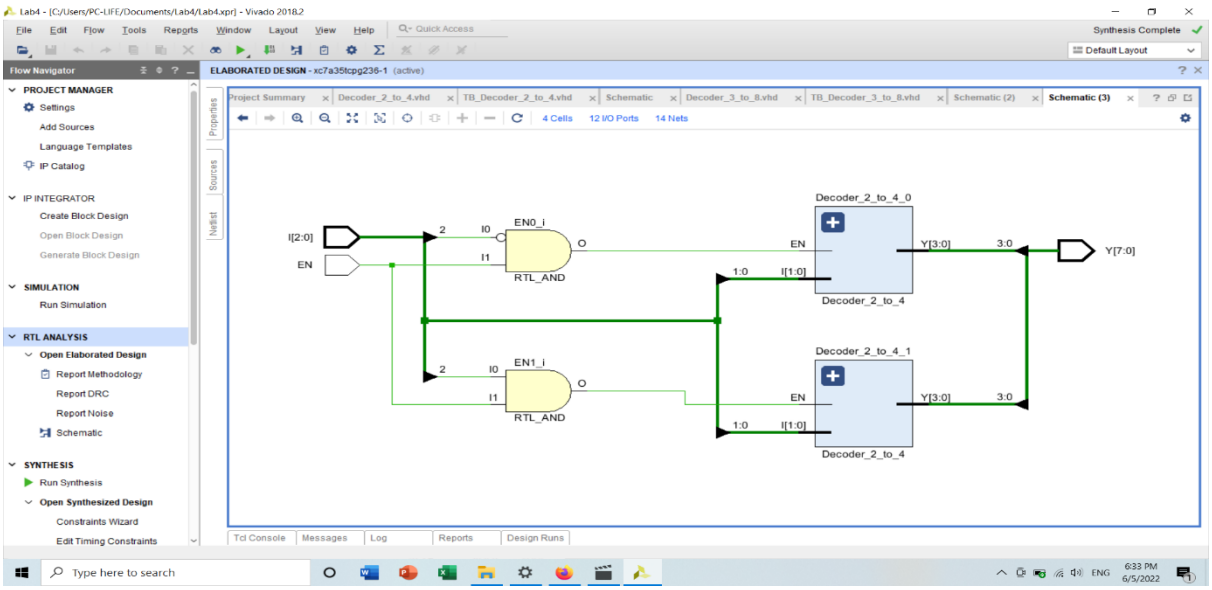


RTL Analysis Elaborated Design Schematics

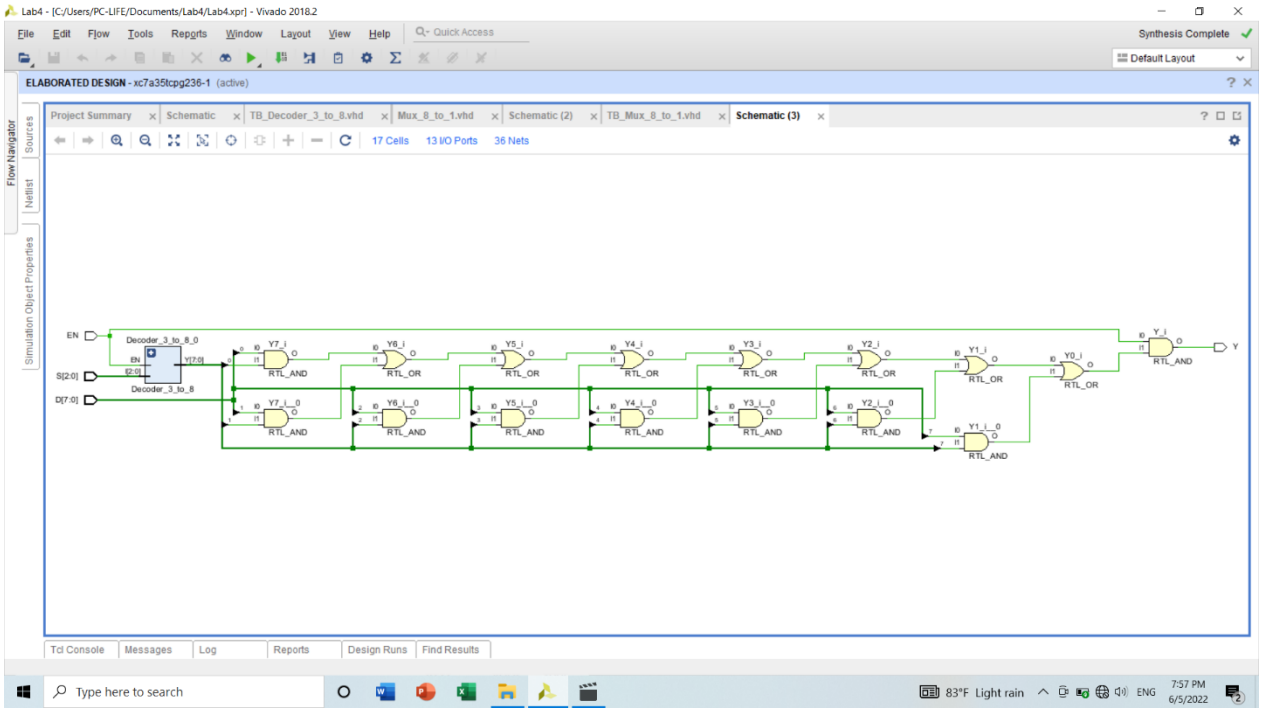
01.RTL Schematic for 2 to 4 Decoder



02.RTL Schematic for 3 to 8 Decoder

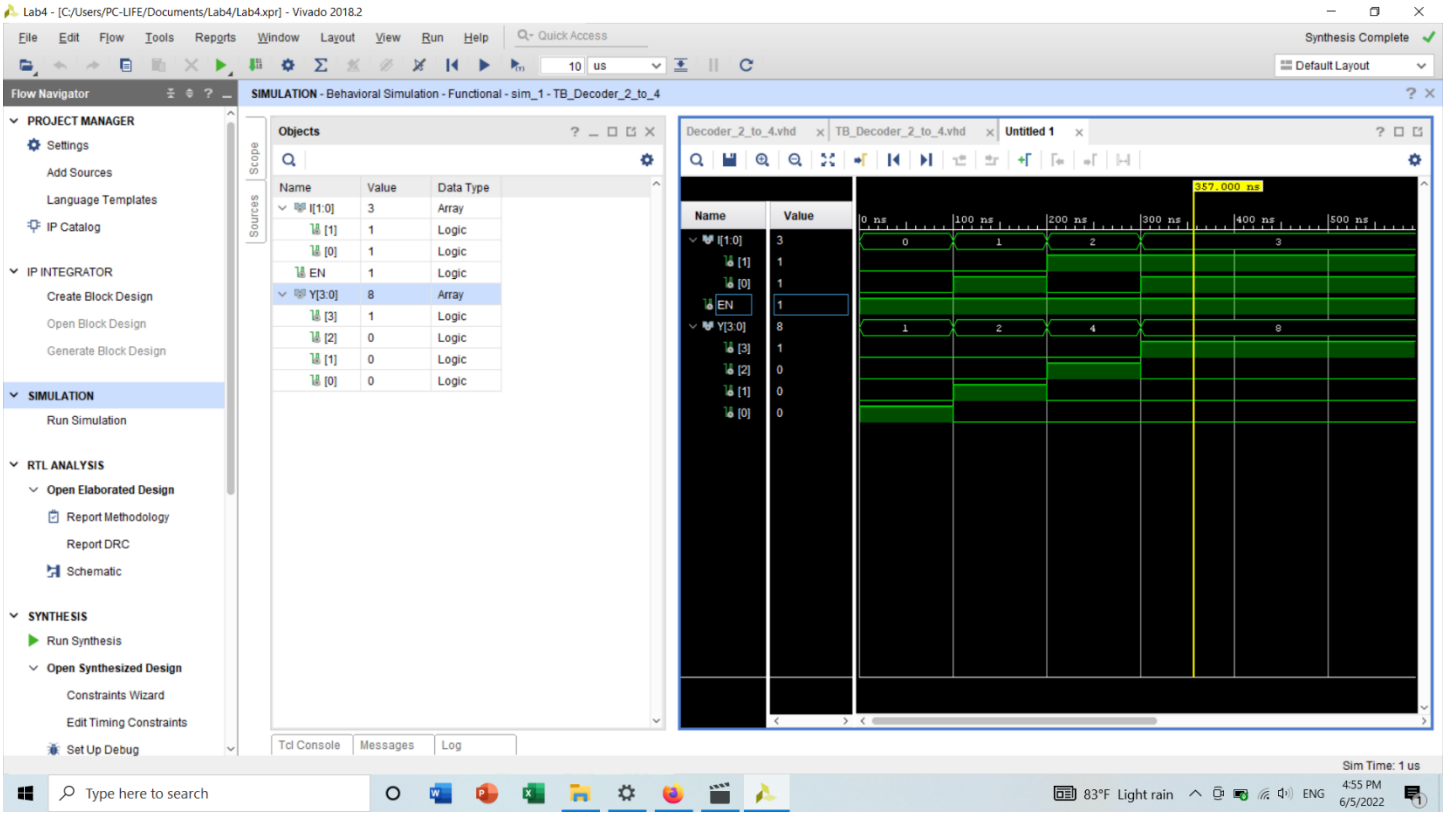


03.RTL Schematic for 8 to 1 Multiplexer

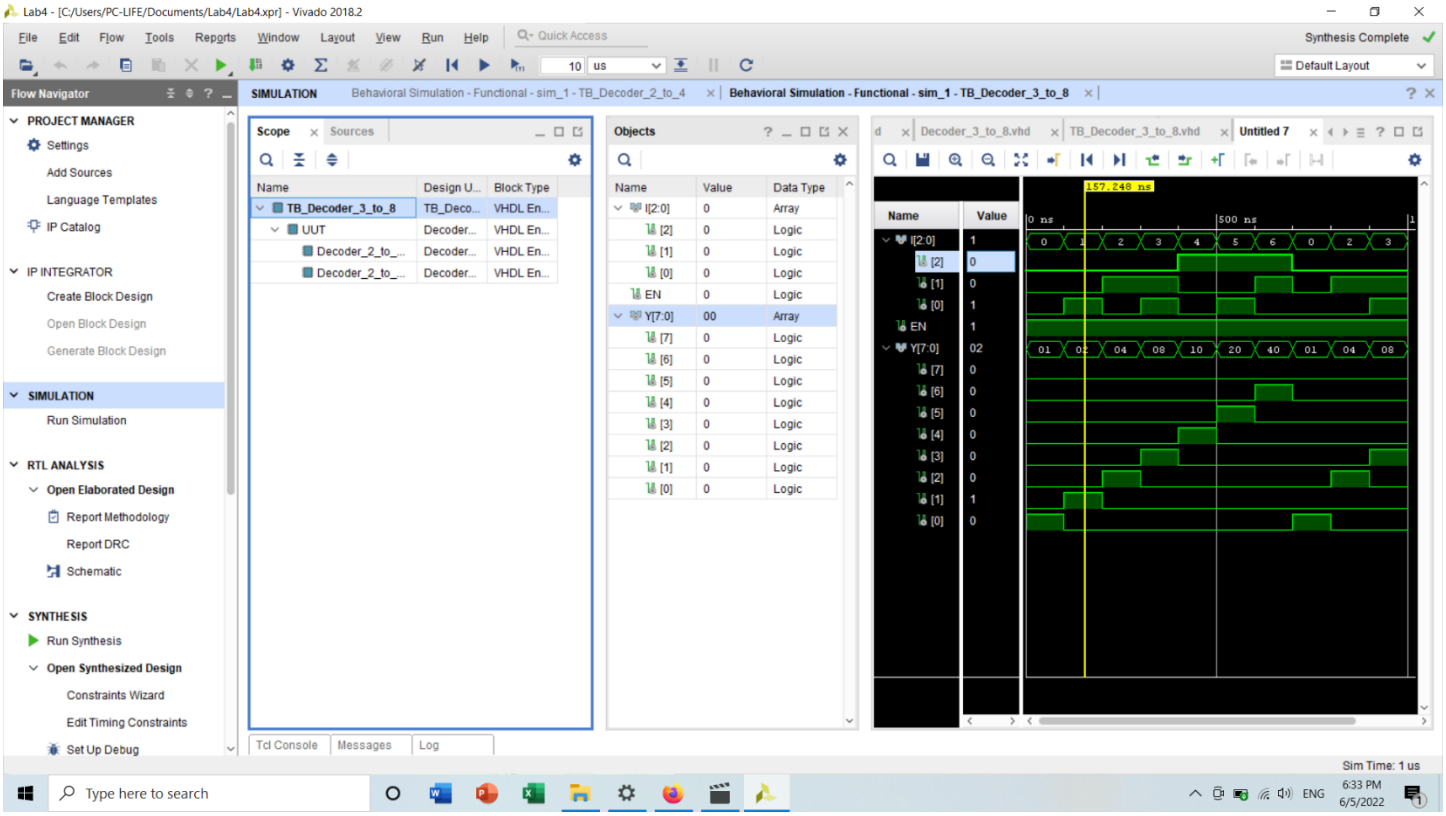


Timing Diagrams

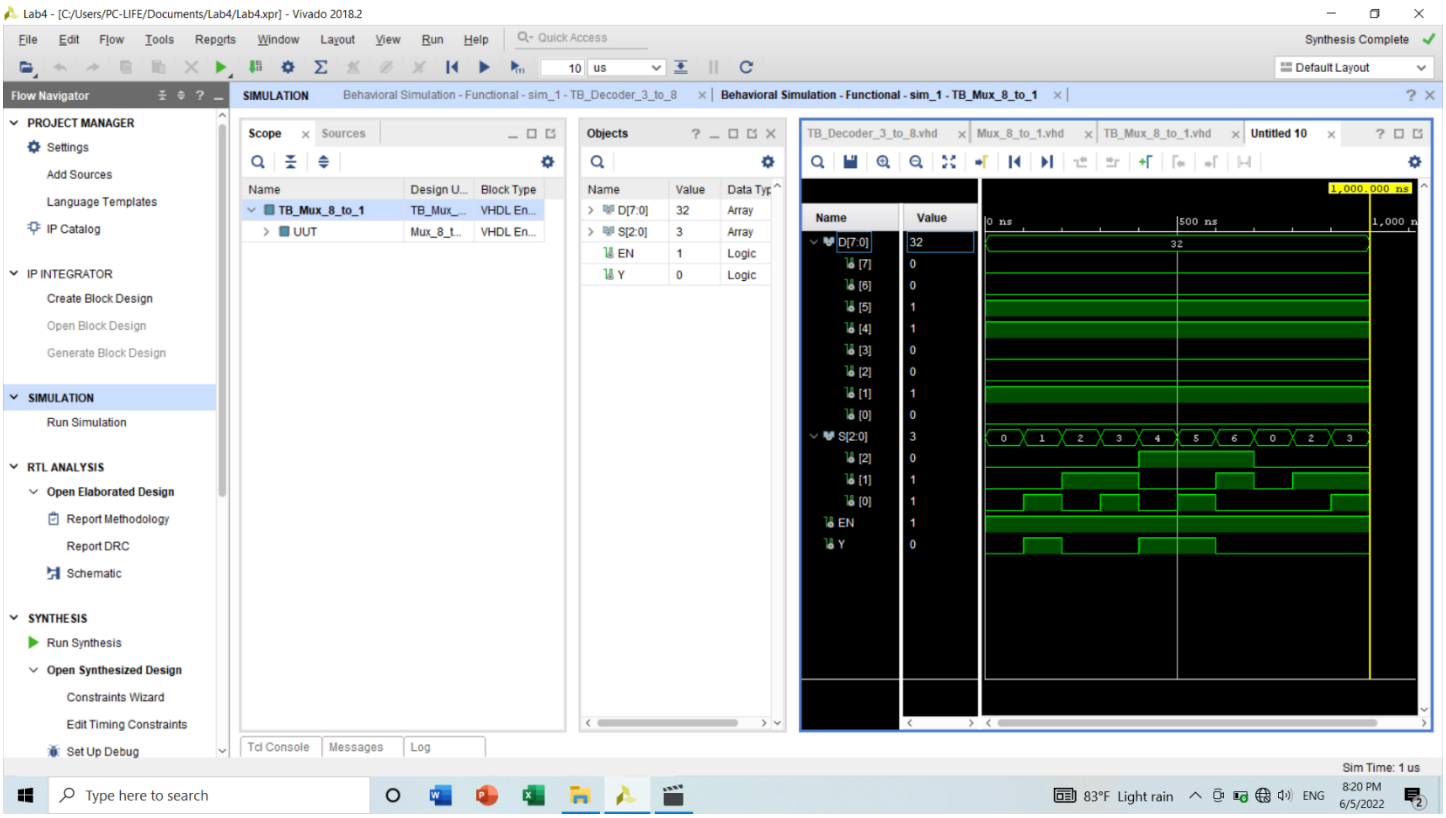
01.Timing Diagram for 2 to 4 decoder



02.Timing Diagram for 3 to 8 decoder

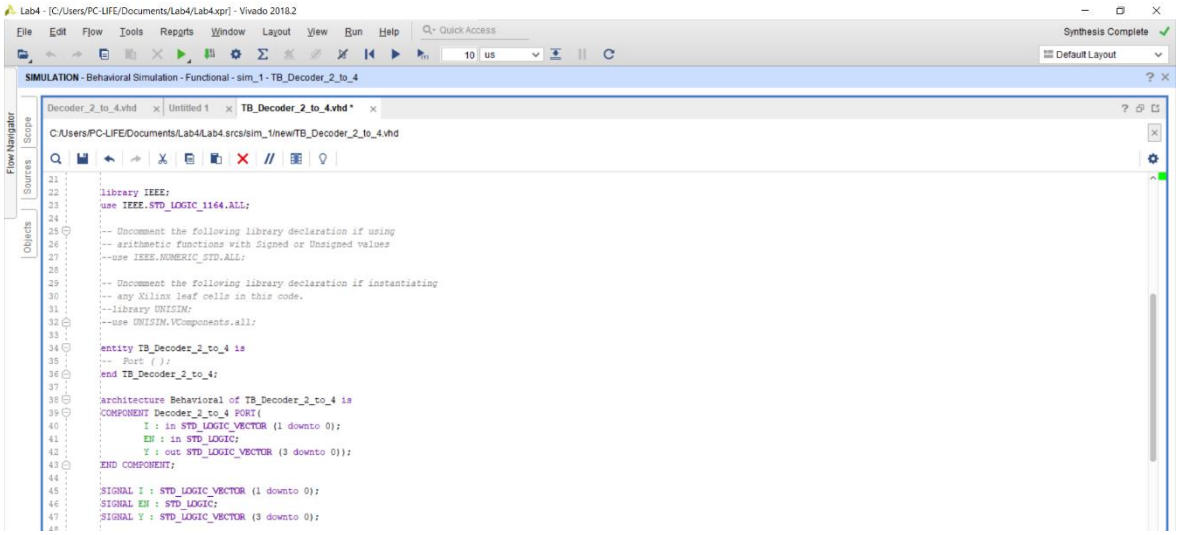


03.Timing Diagram for 8 to 1 multiplexer

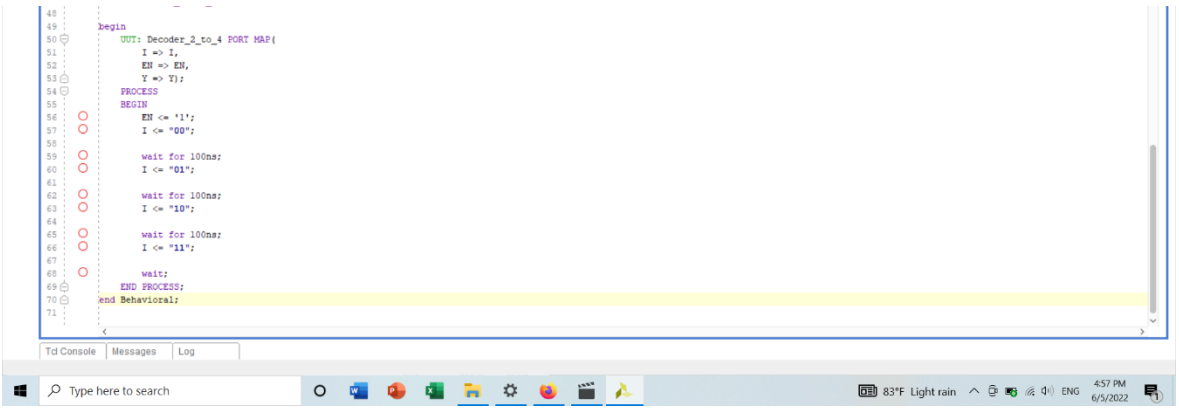


Test Bench Codes

01.Test Bench code for 2 to 4 decoder's simulation

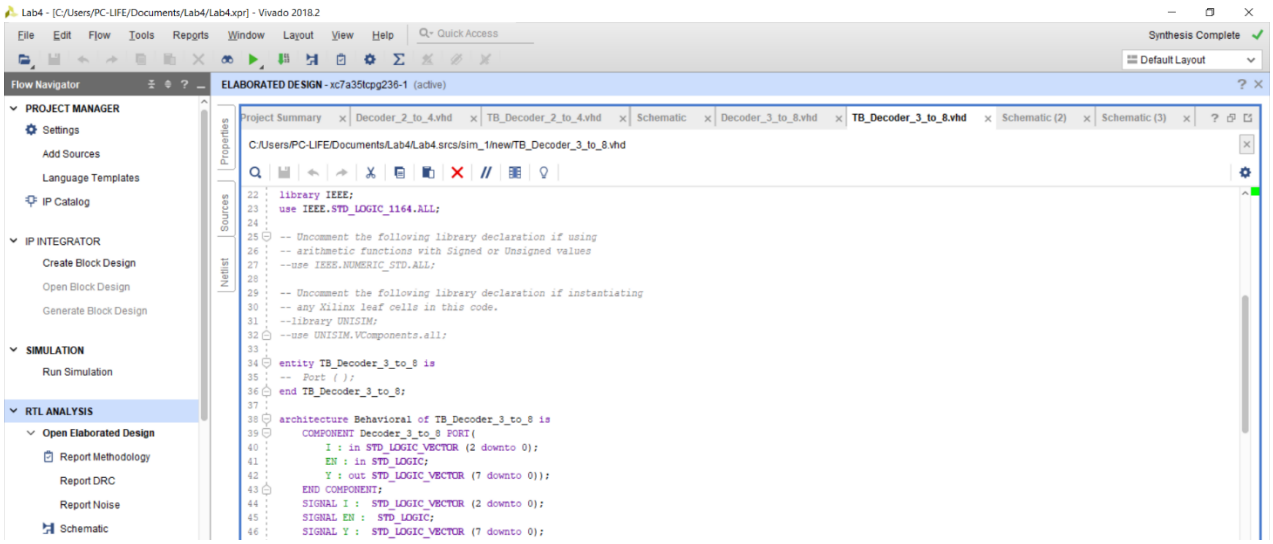


```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity TB_Decoder_2_to_4 is
--  Port ( );
end TB_Decoder_2_to_4;

architecture Behavioral of TB_Decoder_2_to_4 is
COMPONENT Decoder_2_to_4 PORT(
        I : in STD_LOGIC_VECTOR (1 downto 0);
        EN : in STD_LOGIC;
        Y : out STD_LOGIC_VECTOR (3 downto 0));
END COMPONENT;

SIGNAL I : STD_LOGIC_VECTOR (1 downto 0);
SIGNAL EN : STD_LOGIC;
SIGNAL Y : STD_LOGIC_VECTOR (3 downto 0);
```

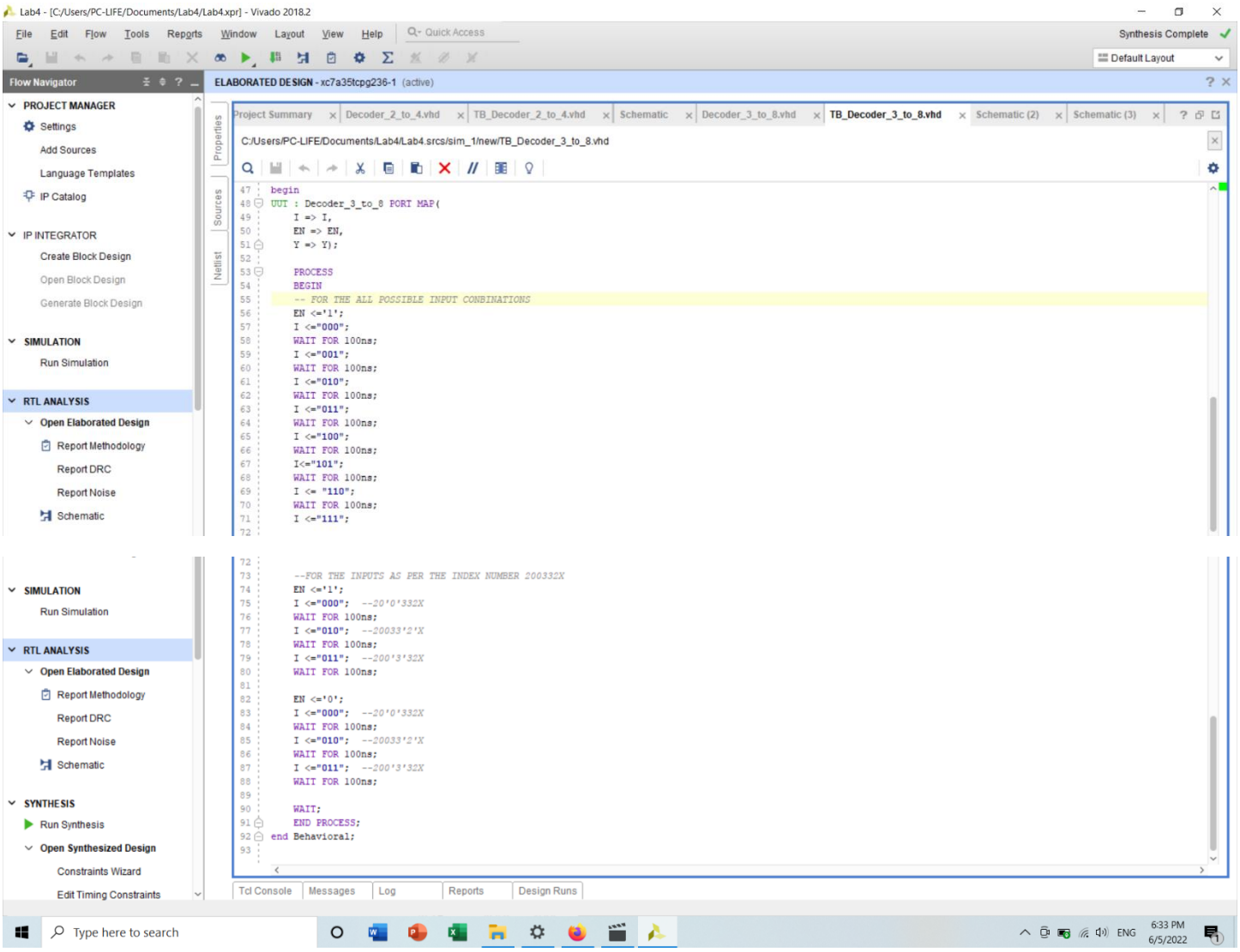


```
begin
    UUT: Decoder_2_to_4 PORT MAP(
        I => I,
        EN => EN,
        Y => Y);
    PROCESS
    BEGIN
        EN <= '1';
        I <= "00";

        wait for 100ns;
        I <= "01";

        wait for 100ns;
        I <= "10";

        wait for 100ns;
        I <= "11";

        wait;
    END PROCESS;
end Behavioral;
```

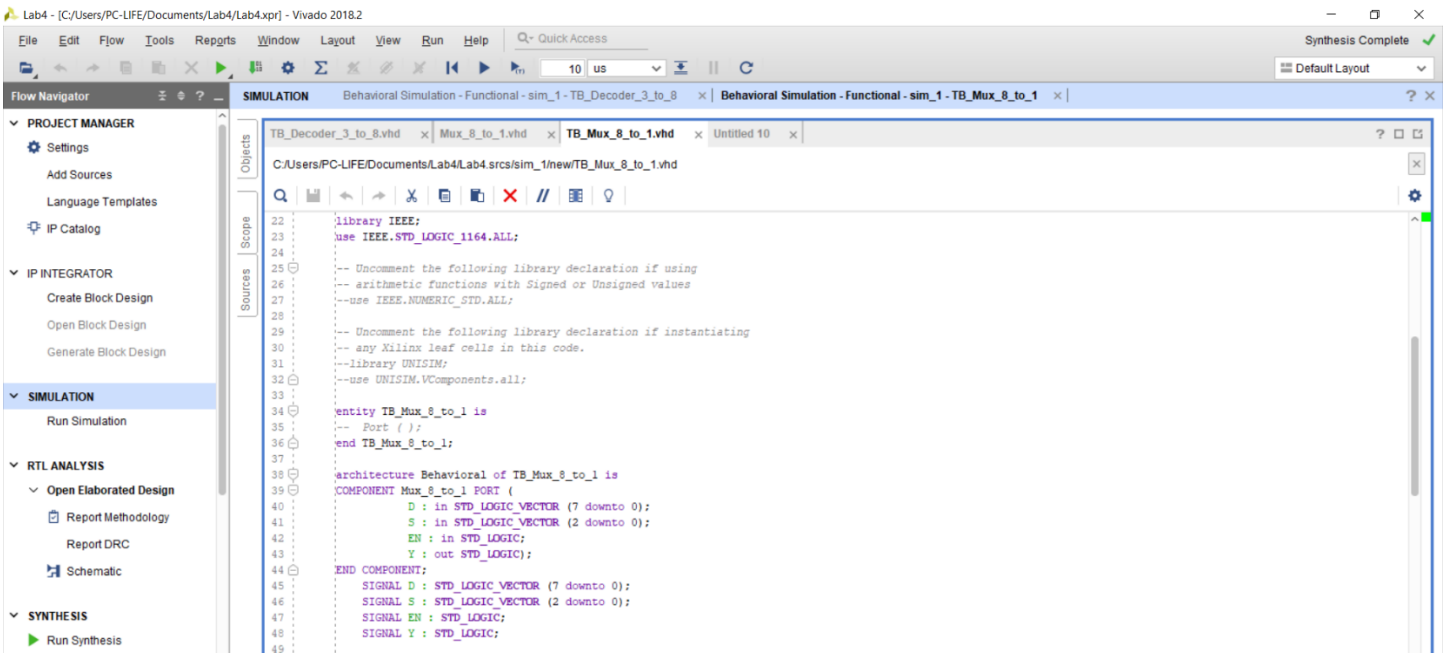
02.Test Bench code for 3 to 8 decoder's simulation

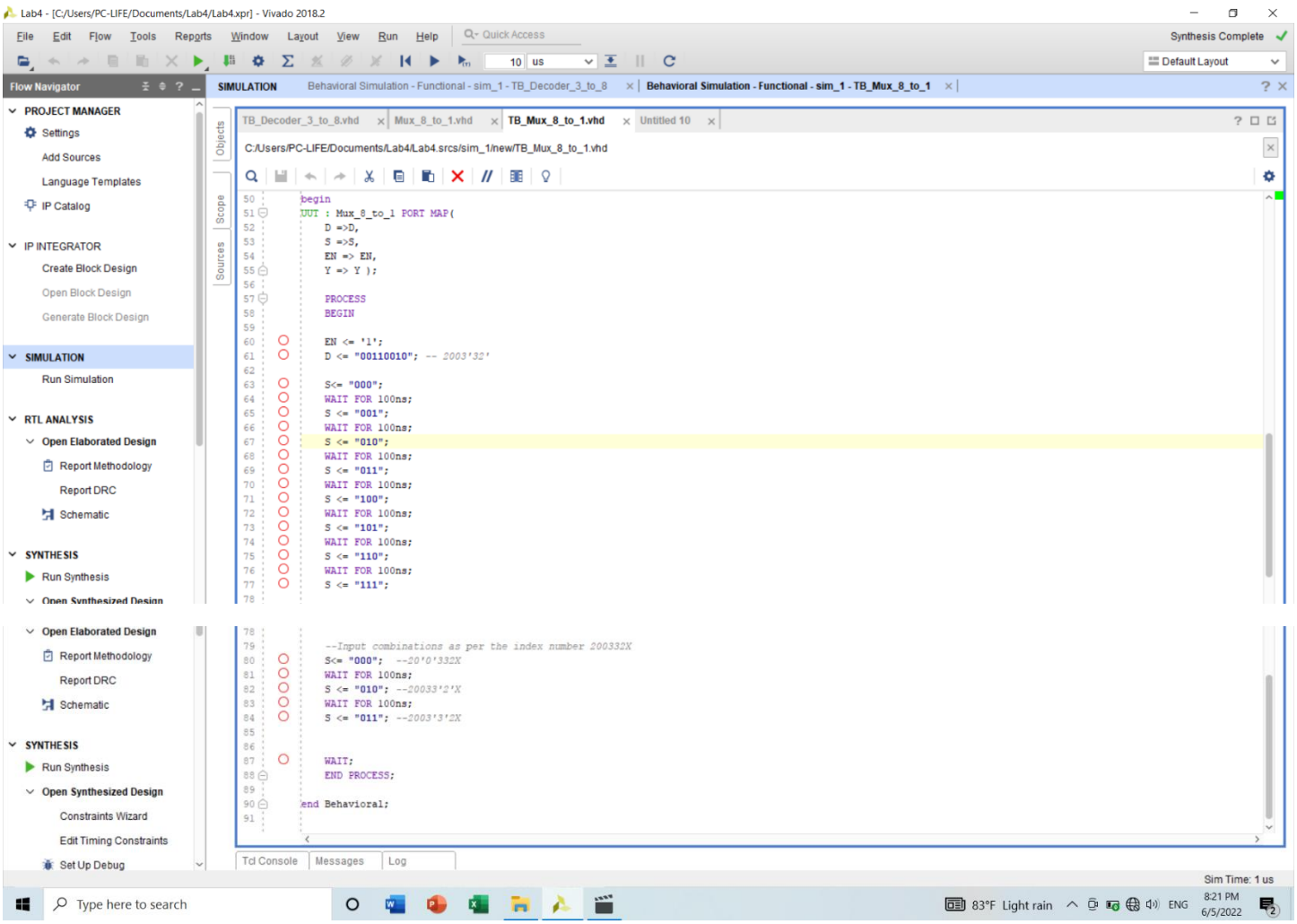


```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity TB_Decoder_3_to_8 is
--  Port ( );
end TB_Decoder_3_to_8;

architecture Behavioral of TB_Decoder_3_to_8 is
    COMPONENT Decoder_3_to_8 PORT(
        I : in STD_LOGIC_VECTOR (2 downto 0);
        EN : in STD_LOGIC;
        Y : out STD_LOGIC_VECTOR (7 downto 0));
    END COMPONENT;
    SIGNAL I :  STD_LOGIC_VECTOR (2 downto 0);
    SIGNAL EN :  STD_LOGIC;
    SIGNAL Y :  STD_LOGIC_VECTOR (7 downto 0);
```



03.Test Bench code for 8 to 1 Multiplexer





Conclusions:

- A decoder converts binary data from n coded inputs to a maximum of 2^n unique outputs. *[In this lab 2 to 4 decoder (2^2) & a 3 to 8 decoder (2^3) is created]*
- A multiplexer receives binary data from 2^n lines and connect them to a single output line based on a given n bit selection. *[In this lab 8 to 1 multiplexer is created ($2^3 \rightarrow 1$)]*
- 3 to 8 decoders can be designed using two 2 to 4 decoders.
- 8 to 1 multiplexer can be designed using a 3 to 8 decoder.
- Functionality of circuits can be verified with a clear visualization using Vivado's simulation process.