Lab 4

Combinational Circuits

CS1050 Computer Organization and Digital Design

RATHNAYAKE R.M.I.B
220526N

Group 41

Task:

As the task, we had to implement a 3-to-8 Decoder and an 8:1 Multiplexer. The implementation of the 3-to-8 decoder was done by using two 2-to-4 Decoders. Then, using this 3-to-8 decoder, we designed and developed an 8-to-1 Multiplexer and ran these simulations for several inputs. Then, finally, we verified the functionality of the 8:1 MUX via simulation and on the BYSYS3 board.

1. 2-to-4 Decoder

INPUT			OUTPUT			
EN	**I(1)**	**I(0)**	**Y(0)**	**Y(1)**	**Y(2)**	**Y(3)**
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Using minterms;

$$Y(0) = \overline{I(0)} \cdot \overline{I(1)}.EN \qquad Y(1) = I(0) \cdot \overline{I(1)}.EN$$

$$Y(2) = \overline{I(0)} \cdot I(1).EN \qquad Y(3) = I(0) \cdot I(1).EN$$

- **Design source file**

```
--------------------------------------------------------------------------
--------
-- Company:
-- Engineer:
--
-- Create Date: 02/20/2024 01:58:59 PM
-- Design Name:
-- Module Name: Decoder_2_to_4 - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
--------------------------------------------------------------------------
--------


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 Decoder_2_to_4 is
    Port ( I : in STD_LOGIC_VECTOR (1 downto 0);
           EN : in STD_LOGIC;
           Y : out STD_LOGIC_VECTOR (3 downto 0));
end Decoder_2_to_4;

architecture Behavioral of Decoder_2_to_4 is

begin
    Y(0) <= EN AND (NOT I(0)) AND (NOT I(1));
    Y(1) <= EN AND I(0) AND (NOT I(1));
    Y(2) <= EN AND (NOT I(0)) AND  I(1);
    Y(3) <= EN AND I(0) AND I(1);

end Behavioral;
```

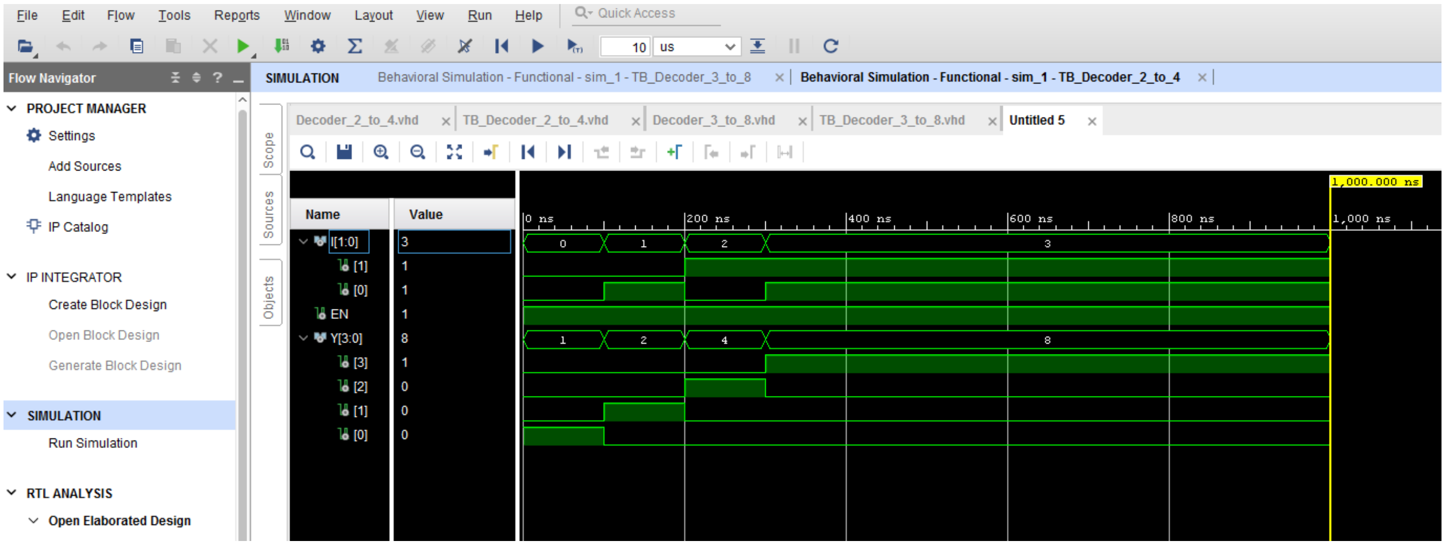
- **Simulation source file**

```
--------------------------------------------------------------------------
--------
-- Company:
-- Engineer:
--
-- Create Date: 02/20/2024 02:01:26 PM
-- Design Name:
-- Module Name: TB_Decoder_2_to_4 - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
--------------------------------------------------------------------------
--------


library IEEE;
```

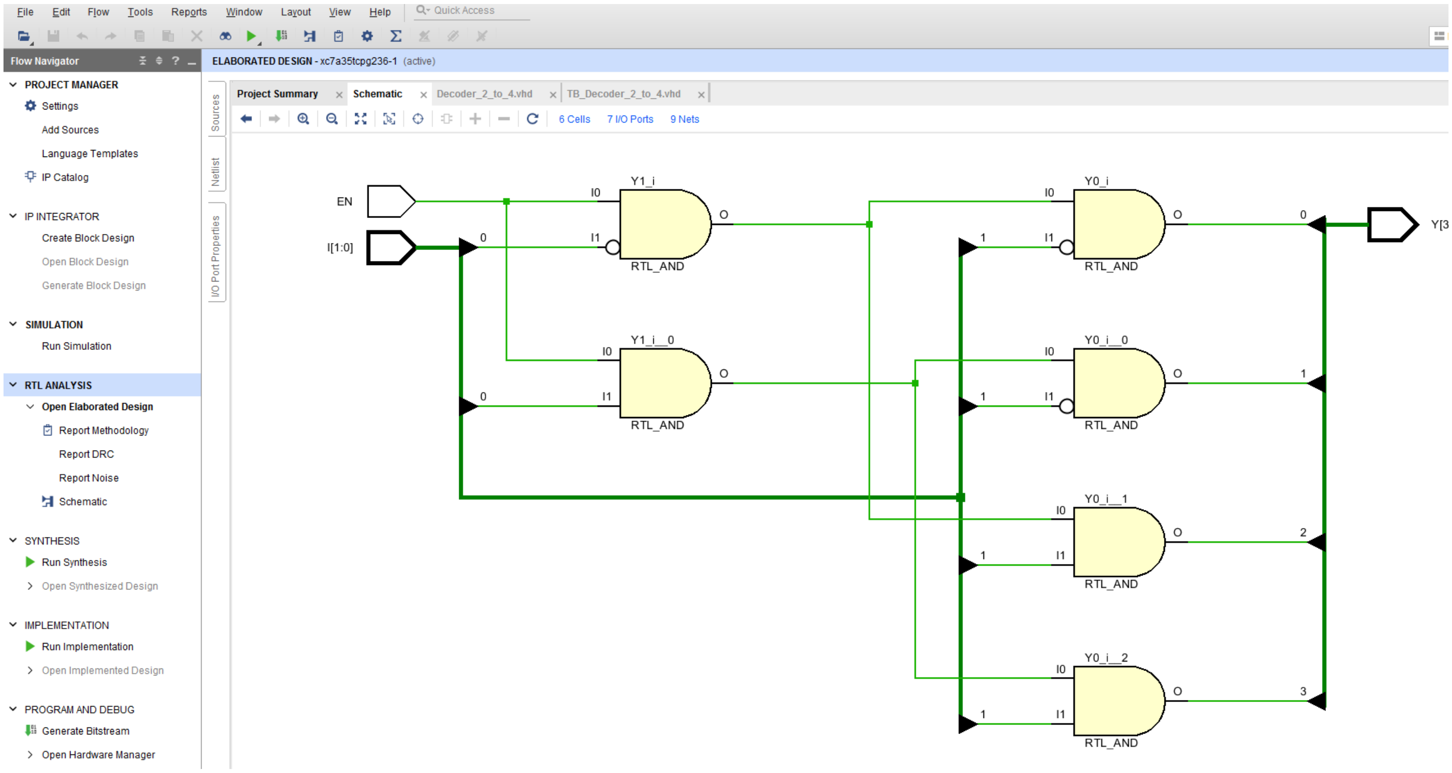
```
use IEEE.STD_LOGIC_1164.ALL;

-- 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

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(0) <= '0';
    I(1) <= '0';
```

```
          WAIT FOR 100 ns ;
          I(0) <= '1';

          WAIT FOR 100 ns ;
          I(1) <= '1';
          I(0) <= '0';

          WAIT FOR 100 ns ;
          I(0) <= '1';

          WAIT;


END PROCESS;

end Behavioral;
```

- **Time diagram**



- **Elaborated design schematic**



2. 3-to-8 Decoder

INPUT				OUTPUT							
EN	I(2)	I(1)	I(0)	Y(7)	Y(6)	Y(5)	Y(4)	Y(3)	Y(2)	Y(1)	Y(0)
0	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

Using minterms;

$Y(0) = EN \cdot \overline{I(0)} \cdot \overline{I(1)} \cdot \overline{I(2)}$ $Y(1) = EN.I(O).\overline{I(1)}.\overline{I(2)}$ $Y(2) = EN.\overline{I(O)}.I(1).\overline{I(2)}$ $Y(3) = EN.I(0).I(1).\overline{I(2)}$

$Y(4) = EN \cdot \overline{I(0)} \cdot \overline{I(1)} \cdot I(2)$ $Y(5) = EN.I(O).\overline{I(1)}.I(2)$ $Y(6) = EN.\overline{I(O)}.I(1).I(2)$ $Y(7) = EN.I(0).I(1).I(2)$

- **Design source file**

```
--------------------------------------------------------------------------
--------
-- Company: 
-- Engineer: 
-- 
-- Create Date: 02/20/2024 03:08:34 PM
-- Design Name: 
-- Module Name: Decoder_3_to_8 - Behavioral
-- Project Name: 
-- Target Devices: 
-- Tool Versions: 
-- Description: 
-- 
-- Dependencies: 
-- 
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
-- 
--------------------------------------------------------------------------
--------
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 Decoder_3_to_8 is
    Port ( I : in STD_LOGIC_VECTOR (2 downto 0);
           EN : in STD_LOGIC;
           Y : out STD_LOGIC_VECTOR (7 downto 0));
end Decoder_3_to_8;
```

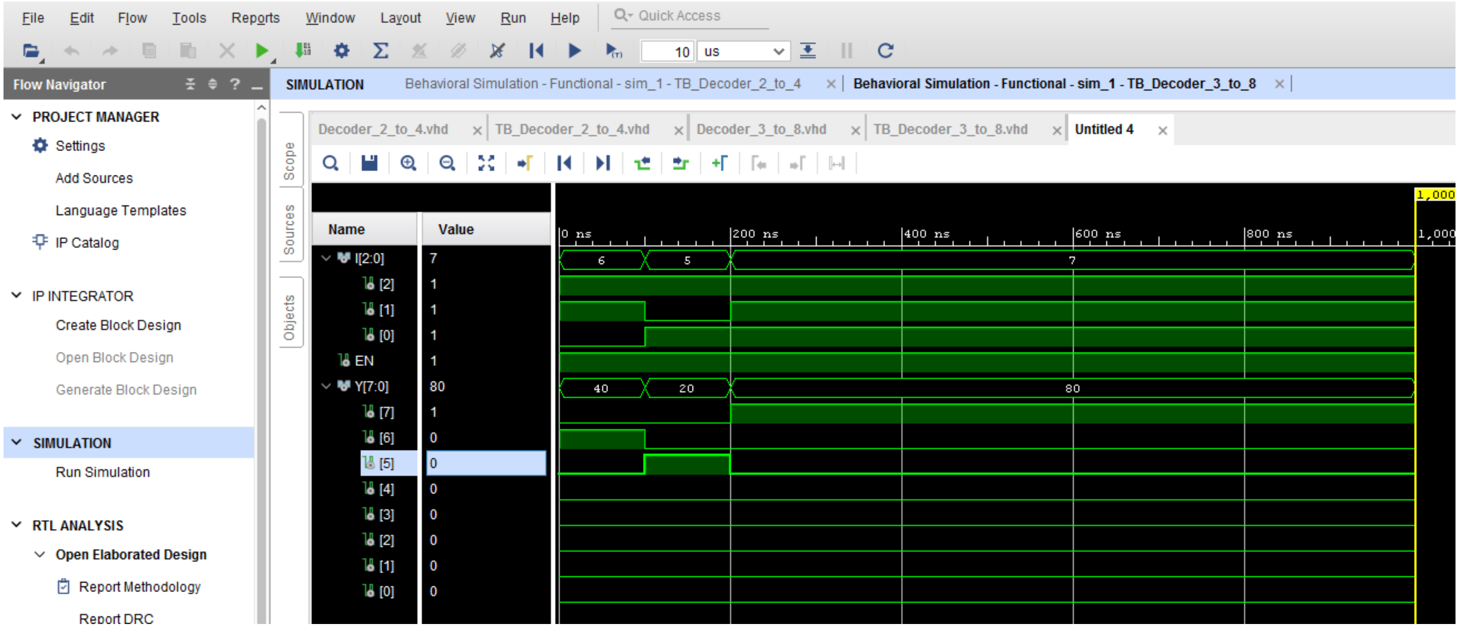
```
architecture Behavioral of Decoder_3_to_8 is

COMPONENT Decoder_2_to_4
    port(
    I: IN STD_LOGIC_VECTOR;
    EN : IN STD_LOGIC;
    Y : OUT STD_LOGIC_VECTOR );
 END COMPONENT;

 SIGNAL I0,I1: STD_LOGIC_VECTOR(1 downto 0);
 SIGNAL Y0,Y1: STD_LOGIC_VECTOR(3 downto 0);
 SIGNAL en0, en1, I2: STD_LOGIC;

begin

    Decode_2_to_4_0 : Decoder_2_to_4
     port map(
     I => I0,
     EN => en0,
     Y => Y0 );

    Decode_2_to_4_1 : Decoder_2_to_4
      port map(
      I => I1,
      EN => en1,
      Y => Y1 );

      en0 <= NOT(I(2)) AND EN;
      en1 <= I(2) AND EN;

      I0 <= I(1 downto 0);
      I1 <= I (1 downto 0);
      I2 <= I(2);

      Y(3 downto 0) <= Y0;
      Y(7 downto 4) <= Y1;

end Behavioral;
```

- **Simulation source file**

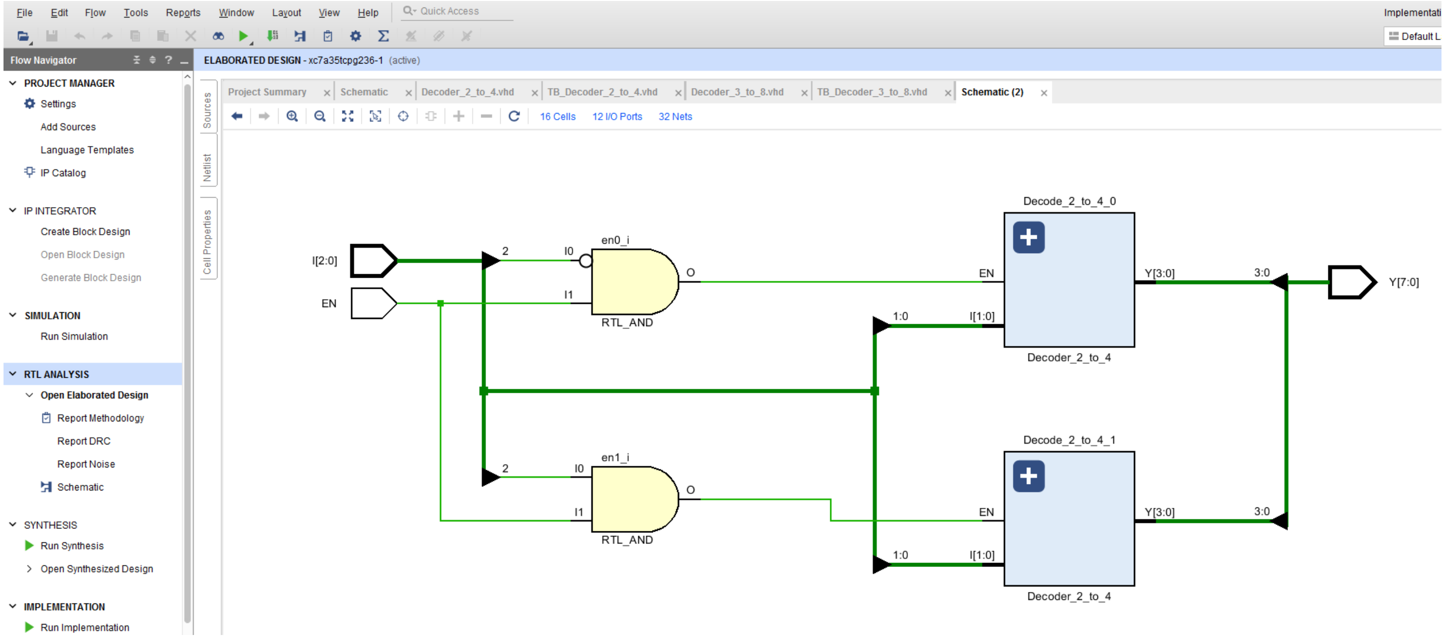
```
--------------------------------------------------------------------------
--------
-- Company:
-- Engineer:
--
-- Create Date: 02/20/2024 03:57:51 PM
-- Design Name:
-- Module Name: TB_Decoder_3_to_8 - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
--------------------------------------------------------------------------
--------
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;
    EN : IN STD_LOGIC;
    Y : OUT STD_LOGIC_VECTOR );
 END COMPONENT;

SIGNAL I : STD_LOGIC_VECTOR (2 downto 0);
SIGNAL EN : STD_LOGIC;
SIGNAL Y : STD_LOGIC_VECTOR (7 downto 0);

begin
UUT: Decoder_3_to_8 PORT MAP(
    I => I,
    EN => EN,
    Y => Y);

PROCESS
BEGIN

    EN <= '1';
    I(0) <= '0';
    I(1) <= '1';
    I(2) <= '1';

    WAIT FOR 100 ns;

    I(0) <= '1';
    I(1) <= '0';
    I(2) <= '1';

    WAIT FOR 100 ns;

    I(0) <= '1';
    I(1) <= '1';
    I(2) <= '1';

    WAIT;
end process;
end Behavioral;
```

- **Time diagram**



- **Elaborated design schematic**



3. 8-to-1 Multiplexer

INPUT												Y
EN	**S(2)**	**S(1)**	**S(0)**	**D(7)**	**D(6)**	**D(5)**	**D(4)**	**D(3)**	**D(2)**	**D(1)**	**D(0)**	
0	X	X	X	X	X	X	X	X	X	X	X	0
1	0	0	0	X	X	X	X	X	X	X	1	1
1	0	0	1	X	X	X	X	X	X	1	X	1
1	0	1	0	X	X	X	X	X	1	X	X	1
1	0	1	1	X	X	X	X	1	X	X	X	1
1	1	0	0	X	X	X	1	X	X	X	X	1
1	1	0	1	X	X	1	X	X	X	X	X	1
1	1	1	0	X	1	X	X	X	X	X	X	1
1	1	1	1	1	X	X	X	X	X	X	X	1

Using minterms;

$$Y = EN.\overline{S(0)}.\overline{S(1)}.\overline{S(2)}.D(0) + EN.S(0).\overline{S(1)}.\overline{S(2)}.D(1) + EN.\overline{S(0)}.S(1).\overline{S(2)}.D(2) + EN.S(0).S(1).\overline{S(2)}.D(3)$$
$$+ EN.\overline{S(0)}.\overline{S(1)}.S(2).D(4) + EN.S(0).\overline{S(1)}.S(2).D(5) + EN.\overline{S(0)}.S(1).S(2).D(6) + EN.S(0).S(1).S(2).D(7)$$

$$Y = EN(\overline{S(0)}.\overline{S(1)}.\overline{S(2)}.D(0) + S(0).\overline{S(1)}.\overline{S(2)}.D(1) + \overline{S(0)}.S(1).\overline{S(2)}.D(2) + S(0).S(1).\overline{S(2)}.D(3) +$$
$$\overline{S(0)}.\overline{S(1)}.S(2).D(4) + S(0).\overline{S(1)}.S(2).D(5) + \overline{S(0)}.S(1).S(2).D(6) + S(0).S(1).S(2).D(7))$$

- **Design source file**

```
------------------------------------------------------------------------------
--------
-- Company:
-- Engineer:
--
-- Create Date: 02/20/2024 05:52:41 PM
-- Design Name:
-- Module Name: MUX_8_to_1 - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
```

```
-- Revision 0.01 - File Created
-- Additional Comments:
--
--------------------------------------------------------------------------
--------



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 MUX_8_to_1 is
    Port ( S : in STD_LOGIC_VECTOR (2 downto 0);
           D : in STD_LOGIC_VECTOR (7 downto 0);
           EN : in STD_LOGIC;
           Y : out STD_LOGIC);
end MUX_8_to_1;

architecture Behavioral of MUX_8_to_1 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 A : std_logic_vector(7 downto 0);

begin
Decoder : Decoder_3_to_8
    port map(
    I => S,
    EN => EN,
    Y => A);

Y <= (D(0) AND A(0)) OR (D(1) AND A(1)) OR (D(2) AND A(2)) OR (D(3) AND A(3)) OR (D(4) AND A(4)) OR (D(5) AND A(5)) OR (D(6) AND A(6)) OR (D(7) AND A(7));

end Behavioral;
```

- **Simulation source file**

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 02/21/2024 12:08:31 PM
-- Design Name:
-- Module Name: TB_MUX_8_to_1 - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
```

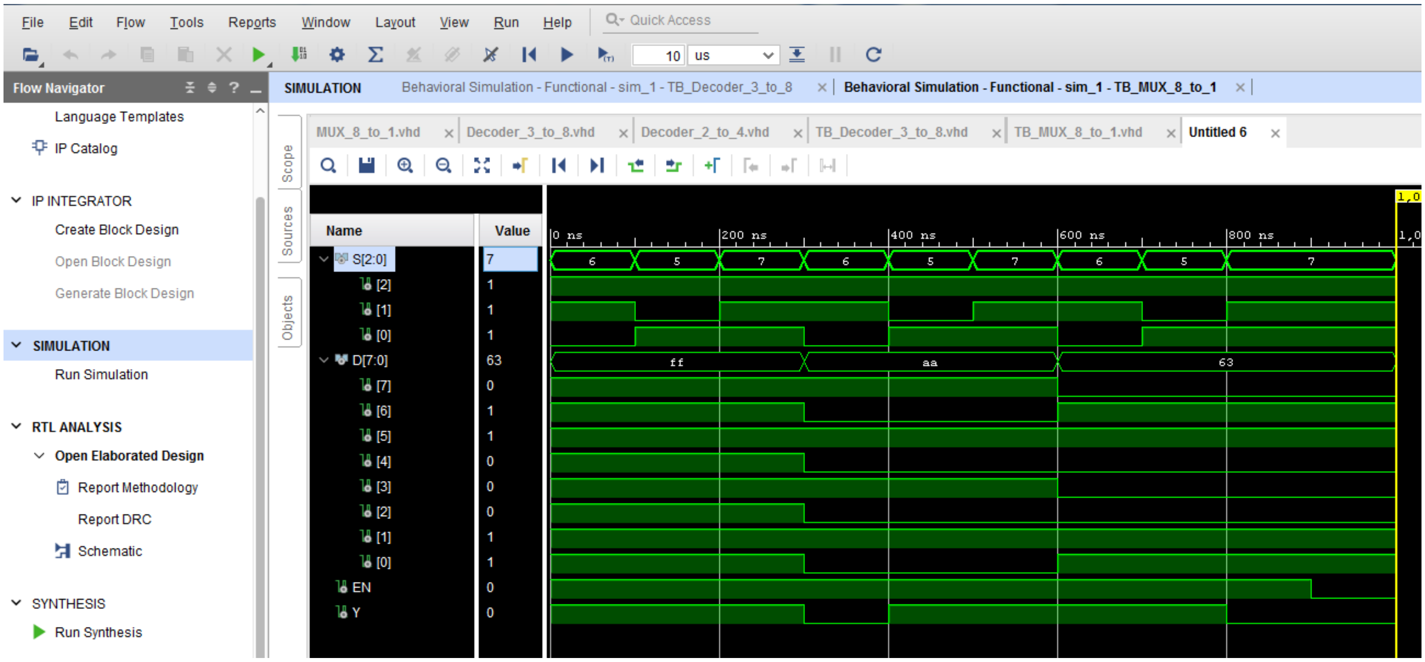
```
--use IEEE.NUMERIC_STD.ALL;

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

entity TB_MUX_8_to_1 is
--  Port ( );
end TB_MUX_8_to_1;

architecture Behavioral of TB_MUX_8_to_1 is
component MUX_8_to_1
    Port (
    S : in STD_LOGIC_VECTOR (2 downto 0);
    D : in STD_LOGIC_VECTOR (7 downto 0);
    EN : in STD_LOGIC;
    Y : out STD_LOGIC);
end component;

signal S : STD_LOGIC_VECTOR (2 downto 0);
signal D : STD_LOGIC_VECTOR (7 downto 0);
signal EN : STD_LOGIC;
signal Y : STD_LOGIC;

begin
UUT: MUX_8_to_1 port map(
    S => S,
    D => D,
    EN => EN,
    Y => Y
);

process
begin

    EN <= '1';
    D <= "11111111";
    S <= "110";
    WAIT FOR 100 ns;
    S <= "101";
    WAIT FOR 100 ns;

    S <= "111";
    WAIT FOR 100 ns;

    D <= "10101010";
    S <= "110";
    WAIT FOR 100 ns;

    S <= "101";
    WAIT FOR 100 ns;

    S <= "111";
    WAIT FOR 100 ns;

    D <= "01100011";
    S <= "110";
    WAIT FOR 100 ns;

    S <= "101";
    WAIT FOR 100 ns;

    S <= "111";
    WAIT FOR 100 ns;

    EN <= '0';

    WAIT ;

end process;
```

- **Constraints file**

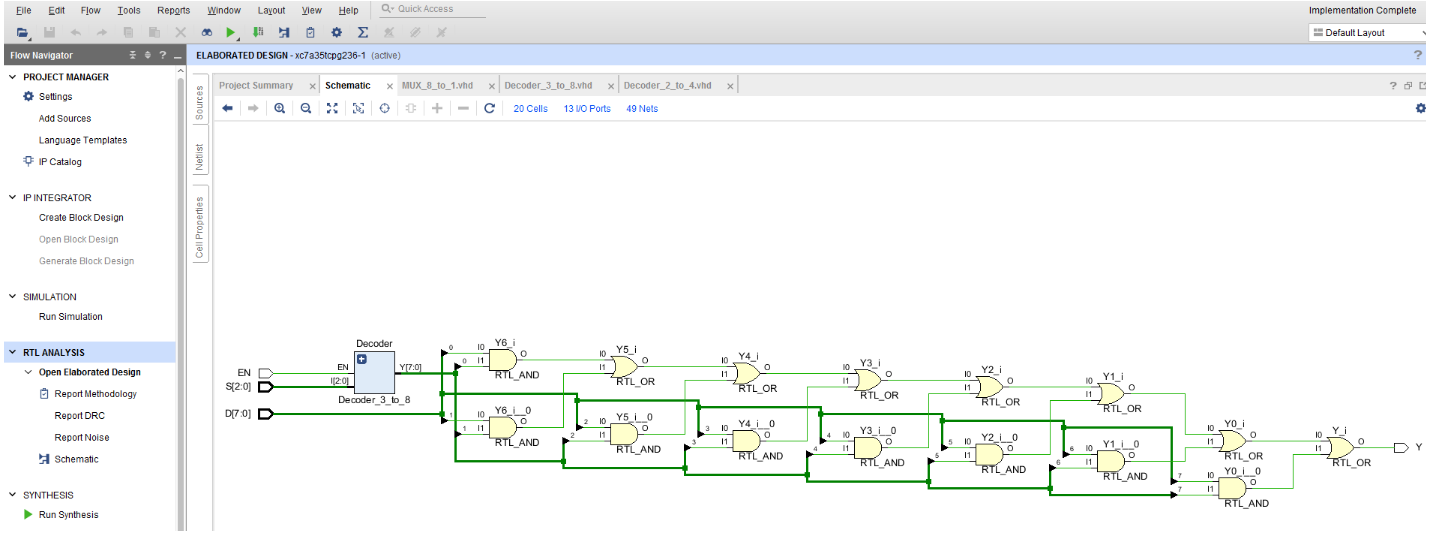
```
## Switches
set_property PACKAGE_PIN V17 [get_ports {D[0]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {D[0]}]
set_property PACKAGE_PIN V16 [get_ports {D[1]}]
```

```
     set_property IOSTANDARD LVCMOS33 [get_ports {D[1]}]
set_property PACKAGE_PIN W16 [get_ports {D[2]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {D[2]}]
set_property PACKAGE_PIN W17 [get_ports {D[3]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {D[3]}]
set_property PACKAGE_PIN W15 [get_ports {D[4]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {D[4]}]
set_property PACKAGE_PIN V15 [get_ports {D[5]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {D[5]}]
set_property PACKAGE_PIN W14 [get_ports {D[6]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {D[6]}]
set_property PACKAGE_PIN W13 [get_ports {D[7]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {D[7]}]
set_property PACKAGE_PIN U1 [get_ports {S[0]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {S[0]}]
set_property PACKAGE_PIN T1 [get_ports {S[1]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {S[1]}]
set_property PACKAGE_PIN R2 [get_ports {S[2]}]
     set_property IOSTANDARD LVCMOS33 [get_ports {S[2]}]

## LEDs
set_property PACKAGE_PIN U16 [get_ports {Y}]
     set_property IOSTANDARD LVCMOS33 [get_ports {Y}]

##Buttons
set_property PACKAGE_PIN U18 [get_ports EN]
     set_property IOSTANDARD LVCMOS33 [get_ports EN]
```

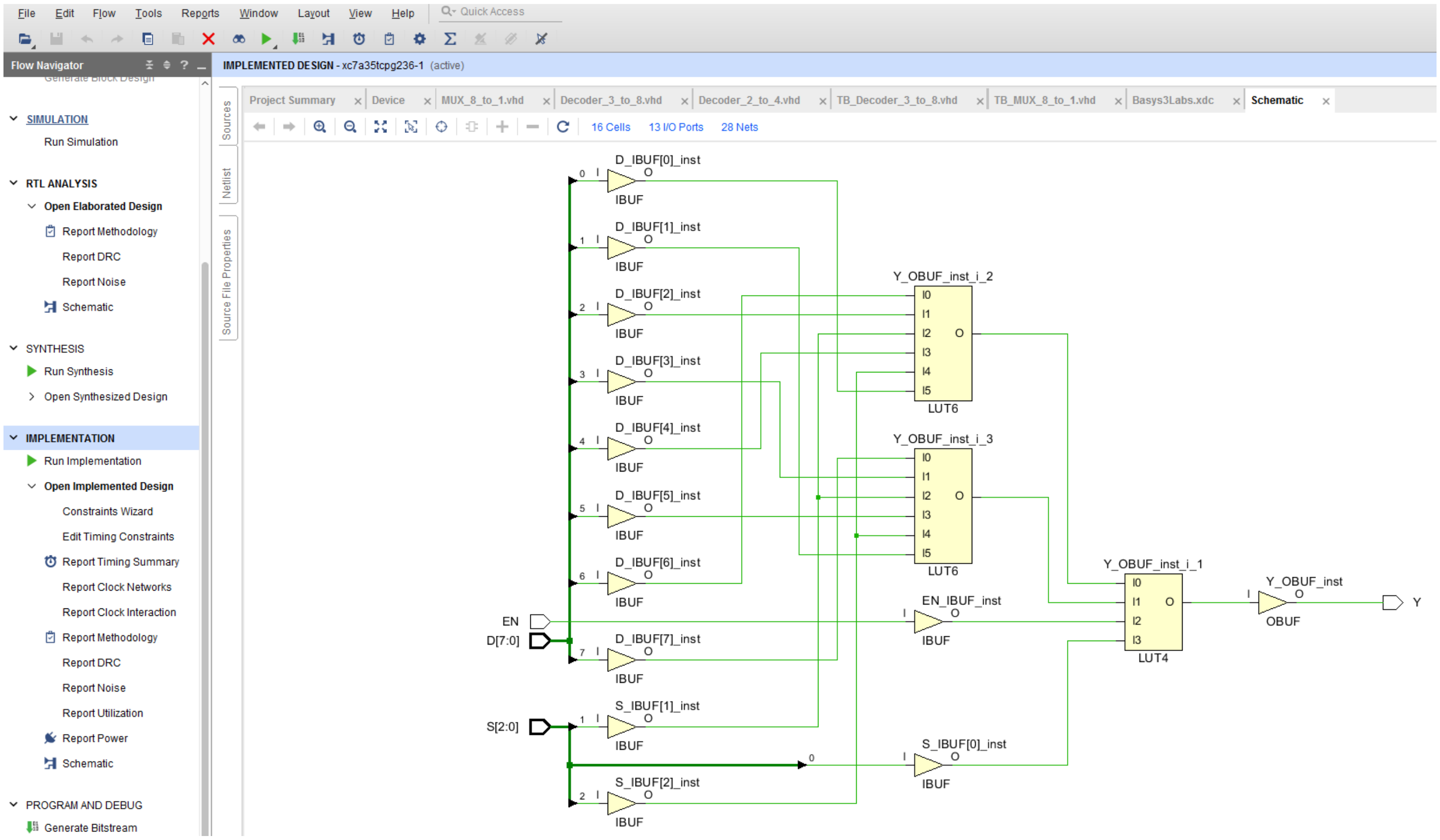
- **Time diagram**



- **Elaborated design schematic**



- **Implemented design schematic**



Conclusion:

Decoders and Multiplexers are two key components of a microprocessor. From n-coded binary input, the decoder can produce a maximum of 2^n distinct outputs. We can easily make a 2-to-4 decoder by using basic logic gates. Then, using two 2-to-4 decoders, we easily designed and developed a 3-to-8 decoder. Providing an “Enable” pin gives the decoders extra functionality (turn on/off).

A multiplexer receives binary data from 2^n lines and connects it to a single output line based on an n-bit selection. We will additionally include an enabling pin in the multiplexer (for turn on/off). Using a control signal, we can control any input to the output. This multiplexer was also built using the pre-built 3-to-8 decoder. This implies that building a complex component using pre-built small components is always much easier. We verified the functionality of the 8:1 MUX via simulation and on the BYSYS3 board.