Lab 4 – Combinational Circuits

CS1050 Computer Organization and Digital Design

Name: Oshadi Saumya Perera

Index no: 200458M

Lab Task

In this lab we will design a decoder and a multiplexer. First we build 2-to-4 decoder. Then we build 3-to-8 decoder by combining two 2-to-4 decoders. Then finally we build 8-to-1 multiplexer using 3-to-8 decoder. We verify their functionality via simulation and on the development board.

VHDL Codes

2-to-4 Decoder design source code (Decoder_2_to_4.vhd)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/02/2022 10:58:48 AM
-- 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(1)) AND NOT(I(0));
Y(1) <= EN AND NOT(I(1)) AND I(0);
Y(2) <= EN AND I(1) AND NOT(I(0));
Y(3) <= EN AND I(1) AND I(0);
end Behavioral;
```

2-to-4 Decoder test bench code (TB_Decoder_2_to_4.vhd)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/02/2022 01:55:28 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
--  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";
      --I(0) <= '0';
      wait for 20ns;
```

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

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

        I <= "11";
        wait;
    end process;

end Behavioral;
```

3-to-8 Decoder design source code (Decoder_3_to_8.vhd)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/02/2022 02:21:59 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 (1 downto 0);
            EN : in STD_LOGIC;
            Y : out STD_LOGIC_VECTOR (3 downto 0)
        );
    end component;
```

```
signal EN0,EN1,I0,I1 : STD_LOGIC;
begin
Decoder_2_to_4_0 : Decoder_2_to_4
PORT MAP(
   I(0) => I0,
   I(1) => I1,
   EN => EN0,
   Y => Y(3 downto 0)
);

Decoder_2_to_4_1 : Decoder_2_to_4
PORT MAP(
   I(0) => I0,
   I(1) => I1,
   EN => EN1,
   Y => Y(7 downto 4)
);
   I0 <= I(0);
   I1 <= I(1);
   EN0 <= NOT(I(2)) AND EN;
   EN1 <= I(2) AND EN;
end Behavioral;
```

3-to-8 Decoder test bench code (TB_Decoder_3_to_8.vhd)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/02/2022 02:44:51 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 (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);
begin
UUT : Decoder_3_to_8
   PORT MAP(
      I => I,
      EN => EN,
      Y => Y
   );
   process
   begin
      EN <= '1';
      I <= "010";
      wait for 20ns;

      I <= "001";
```

```
wait for 20ns;

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

I <= "111";
wait for 20ns;

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

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

EN <= '0';
I <= "010";
wait for 20ns;

I <= "001";
wait for 20ns;

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

I <= "111";
wait for 20ns;

I <= "000";
wait for 20ns;
```

```
        I <= "110";
        wait;

    end process;

end Behavioral;
```

8-to-1 Multiplexer design source code (Mux_8_to_1.vhd)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/02/2022 03:21:18 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 DY :  STD_LOGIC_VECTOR (7 downto 0);
begin
   Decoder_3_to_8_0 : Decoder_3_to_8
      PORT MAP(
         I => S,
         EN => EN,
```

```
        Y => DY
    );
  Y <= EN AND ((D(0) AND DY(0))
            OR (D(1) AND DY(1))
            OR (D(2) AND DY(2))
            OR (D(3) AND DY(3))
            OR (D(4) AND DY(4))
            OR (D(5) AND DY(5))
            OR (D(6) AND DY(6))
            OR (D(7) AND DY(7)));
end Behavioral;
```

8-to-1 Multiplexer test bench code (TB_Mux_8_to_1.vhd)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/02/2022 06:52:52 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
```

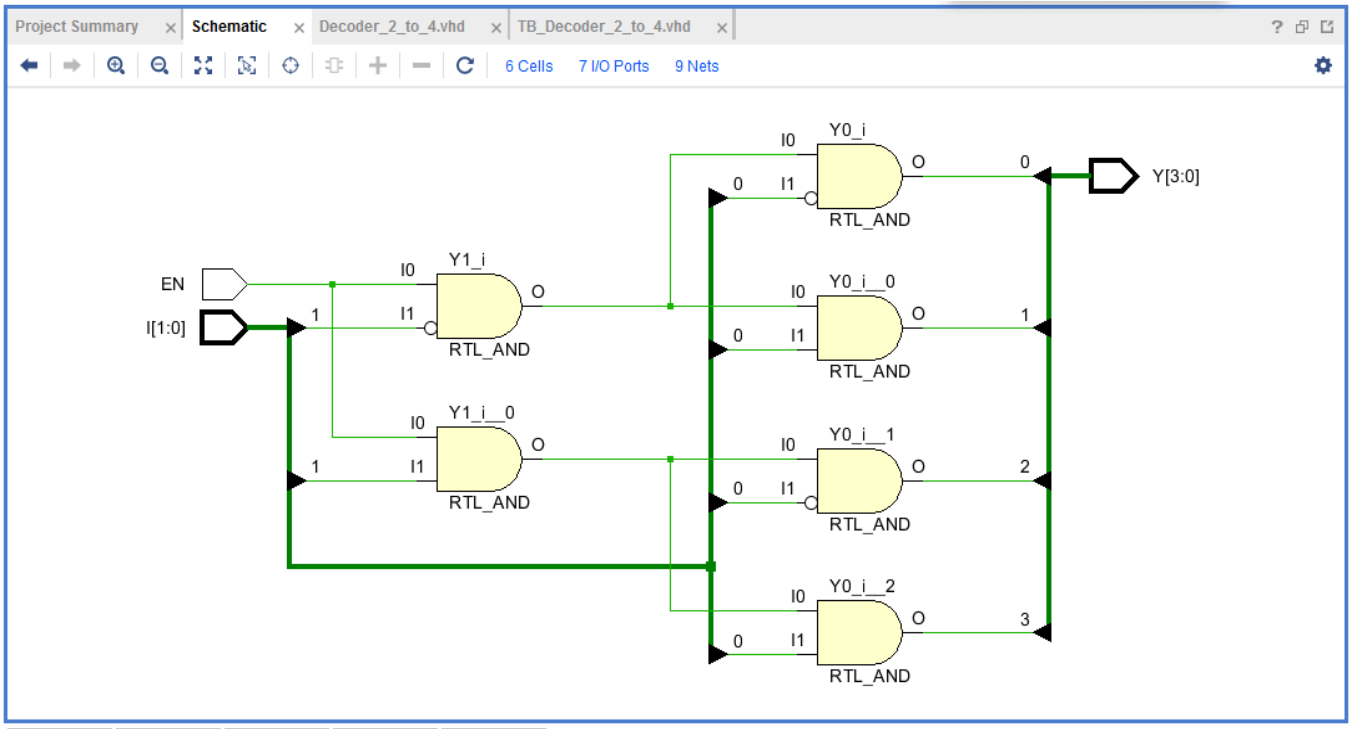
```
-- Additional Comments:
--
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment 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(
        D => D,
        S => S,
        EN => EN,
        Y => Y
    );
    process
    begin
        EN <= '1';
        D <= "01010101";
        S <= "010";

        wait for 20ns;
        S <= "001";

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

        wait for 20ns;
        S <= "111";

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

        wait for 20ns;
```

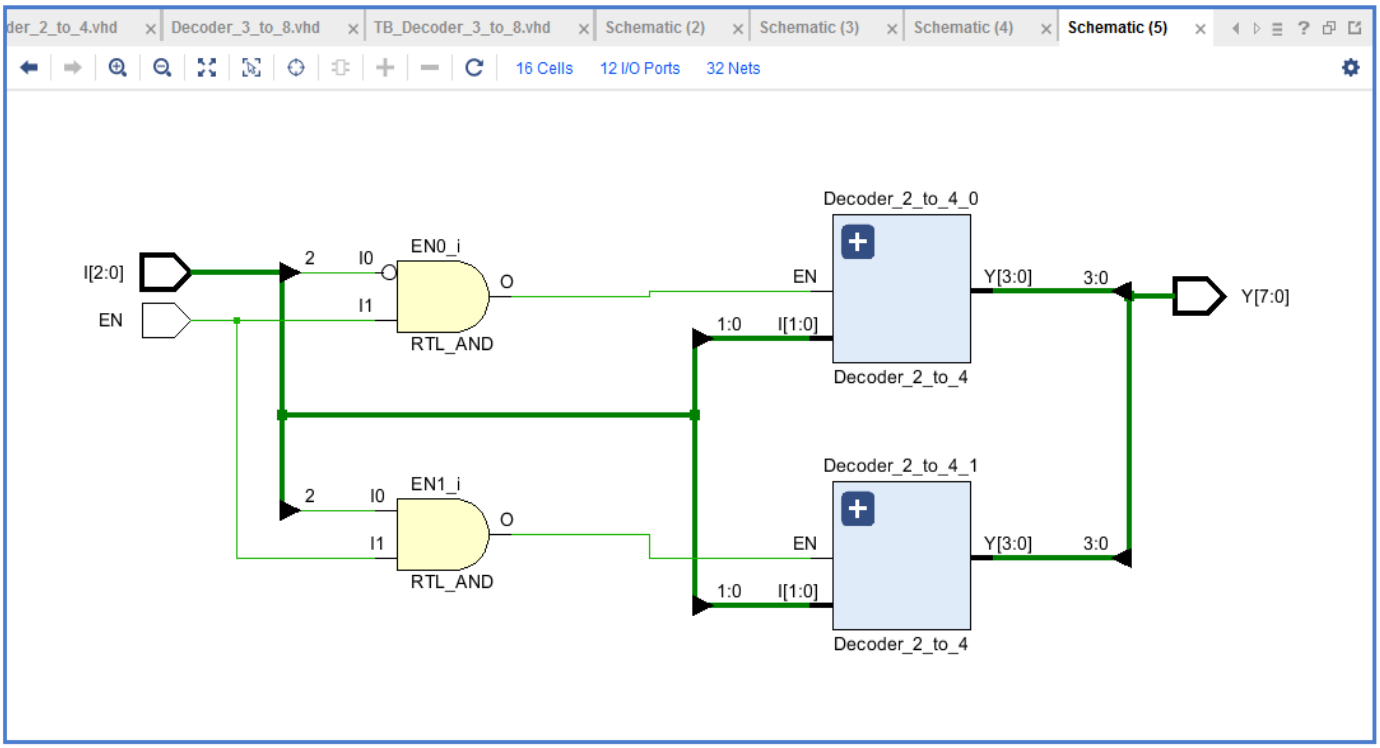
```
        S <= "110";


        wait;
    end process;

end Behavioral;
```

RTL Analysis elaborated design schematics

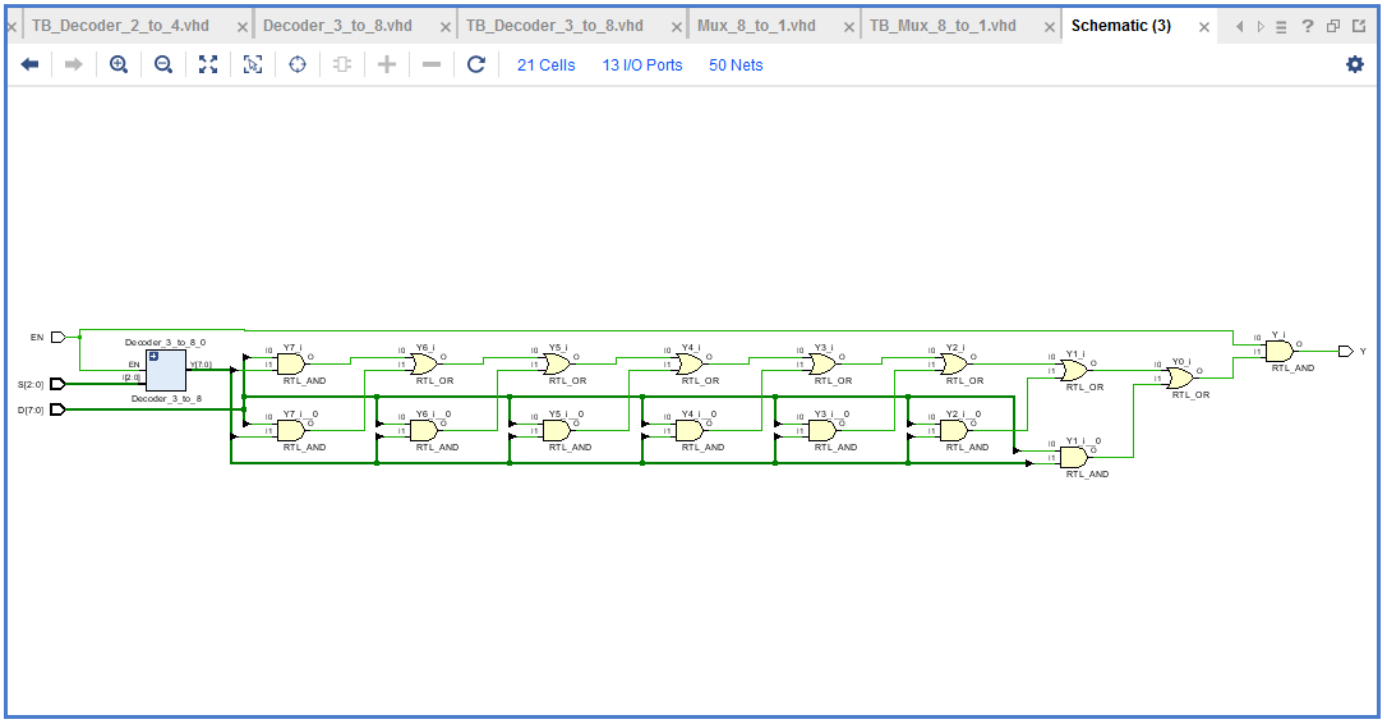
2-to-4 Decoder



3-to-8 Decoder

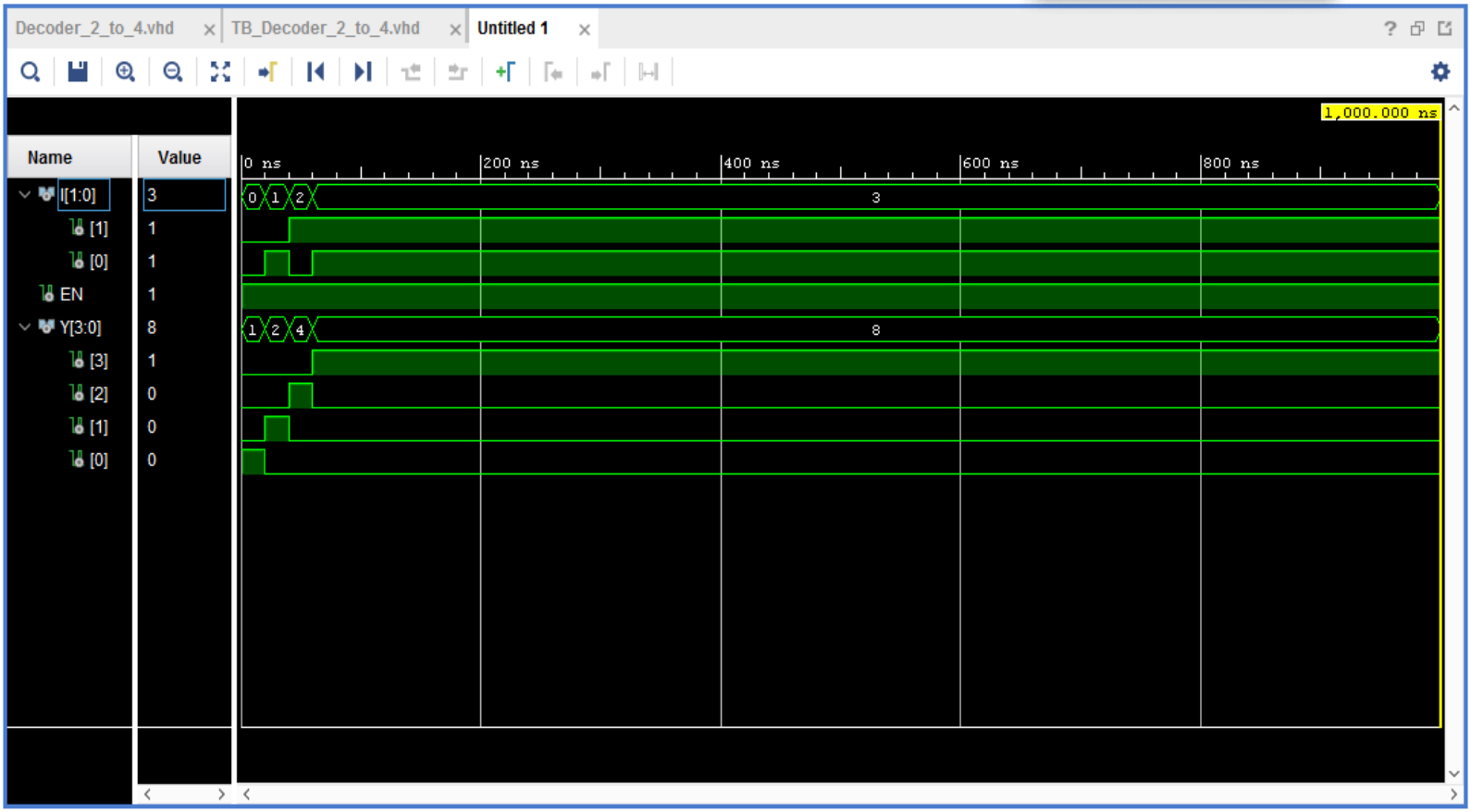


8-to-1 Multiplexer

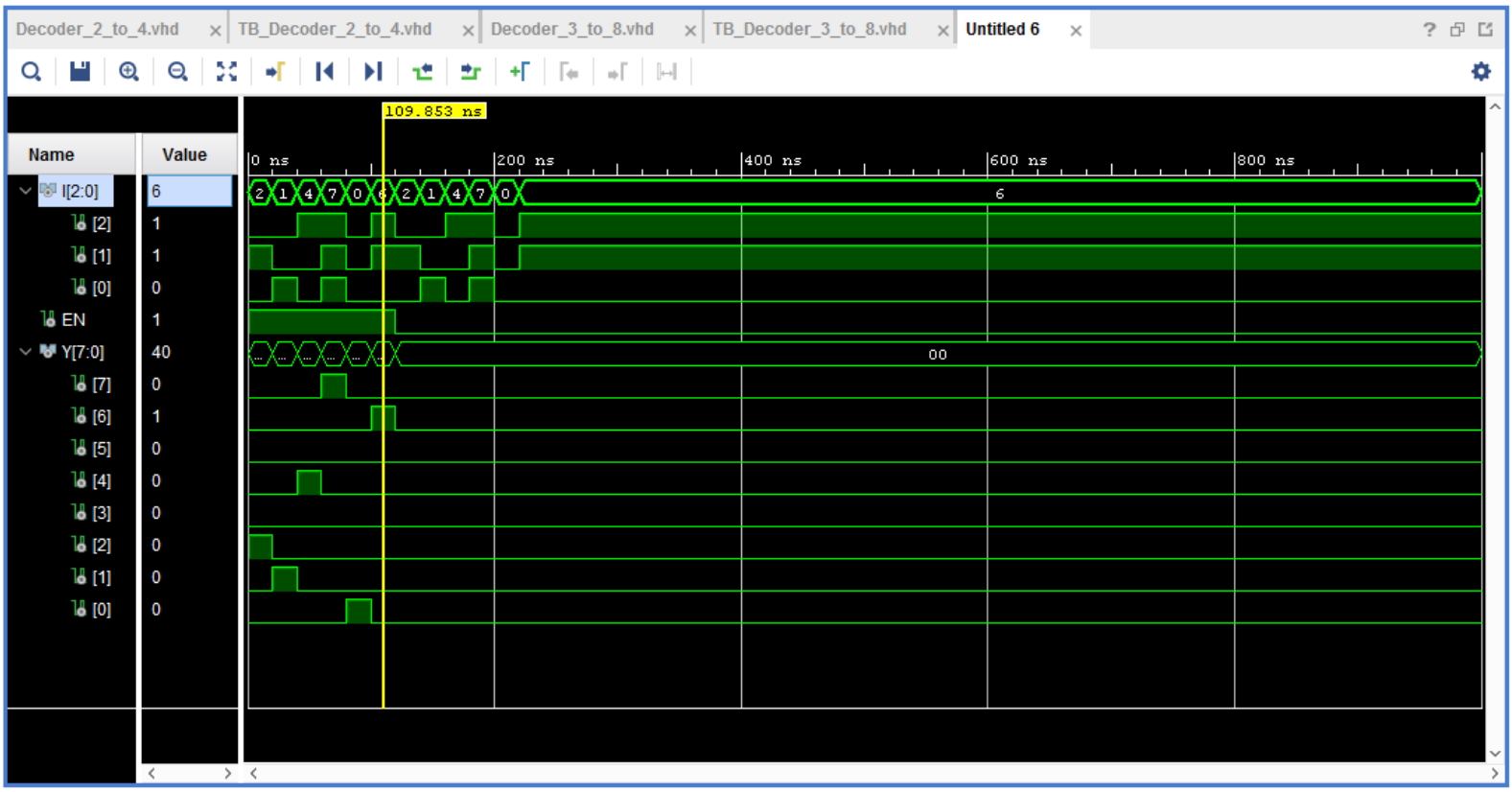


Timing Diagrams

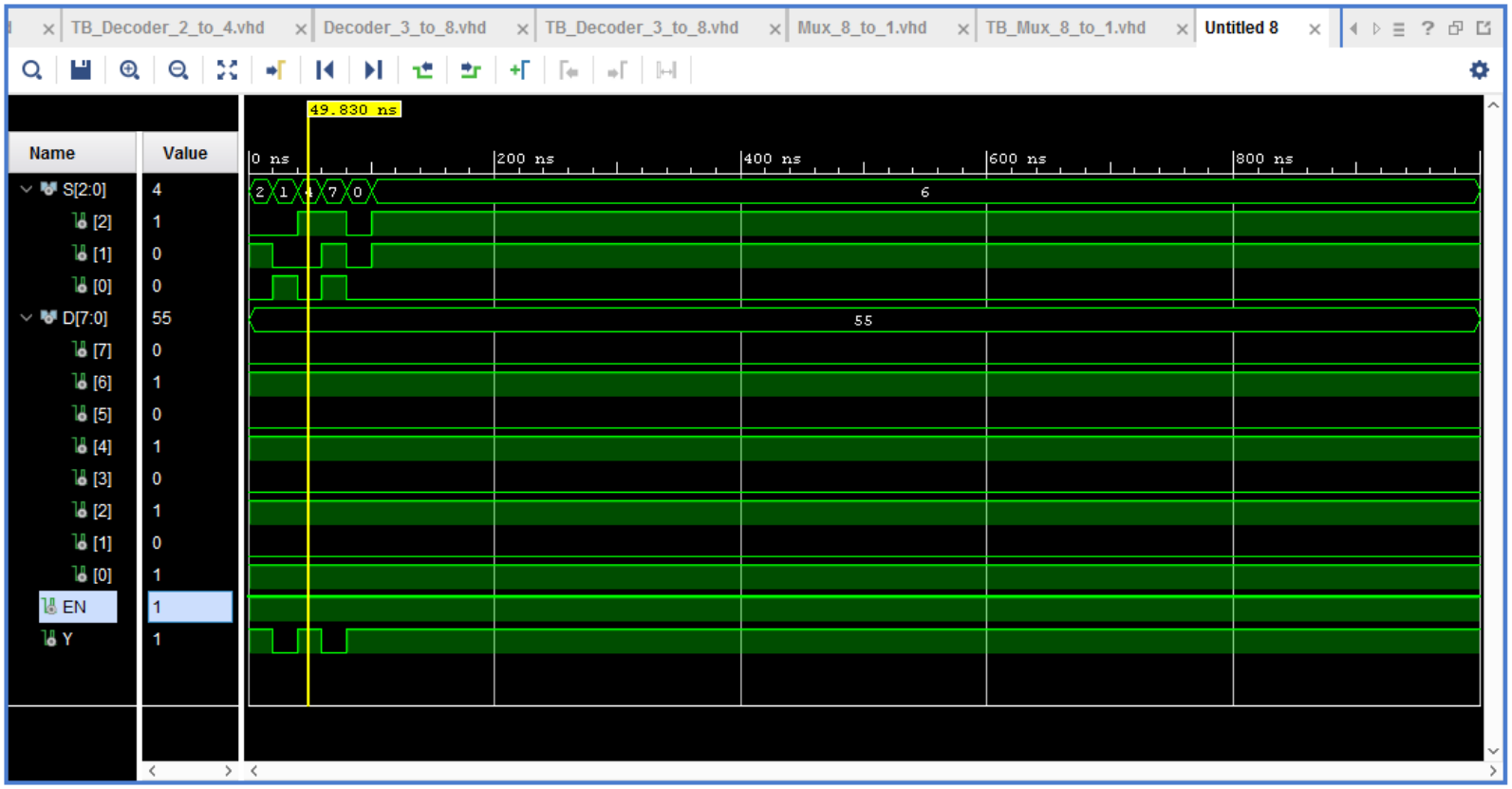
2-to-4 Decoder



3-to-8 Decoder



8-to-1 Multiplexer



Conclusions from the lab

Multiplexers can be used to increase the amount of data that can be transferred through a network. Also decoders and multiplexers are 2 key components of a microprocessor. We can use these components to build a simple microprocessor.