LAB 04 Computer Organization& Design

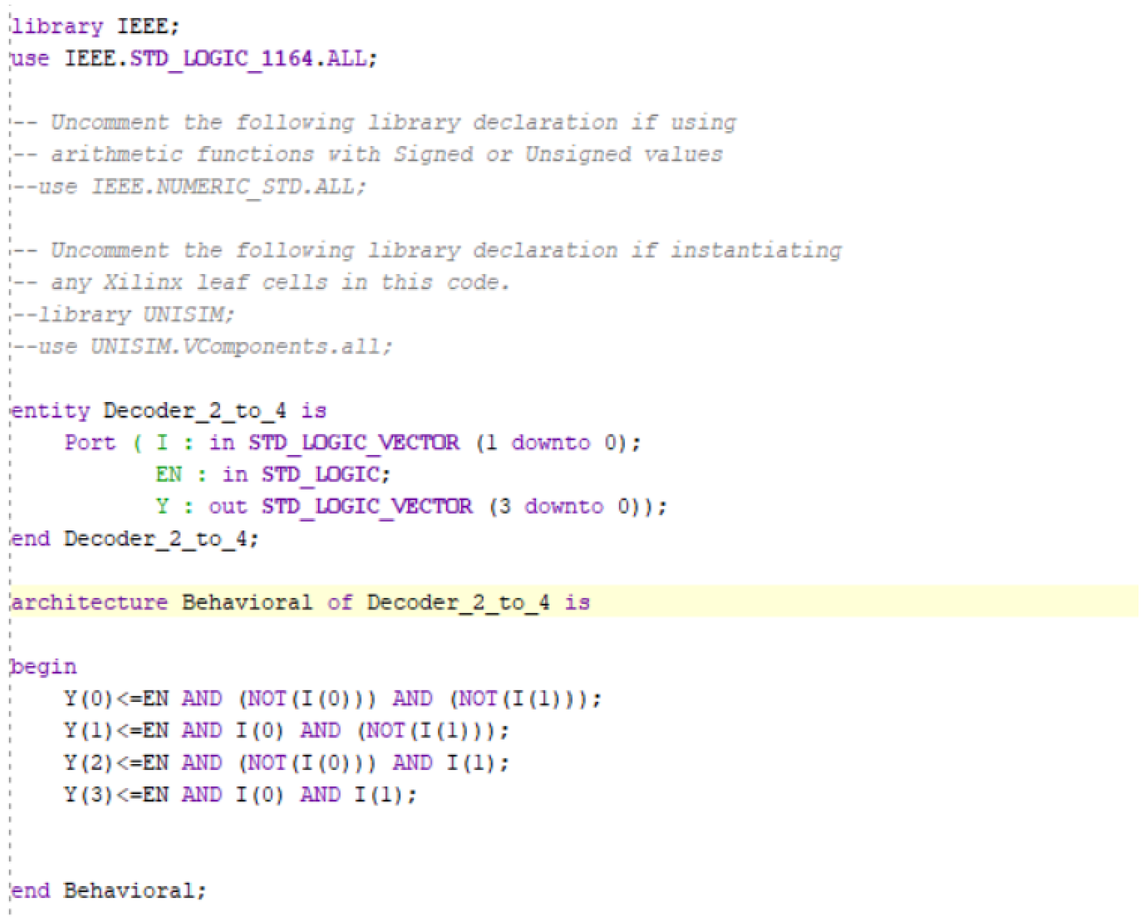
210495G

Nipun Viraj

Assigned Task

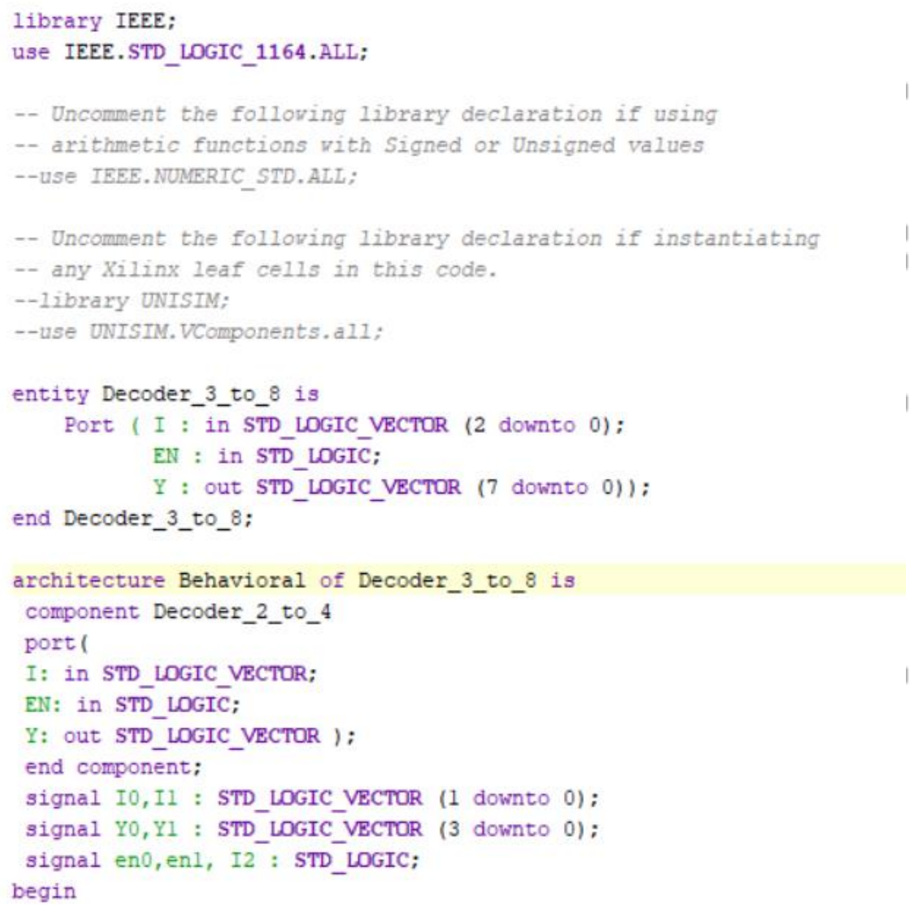
- We have to design a 3-8 decoder using a hierarchical structure with the use of pre-designed 2-4 decoders.
- Also we have to design a 8-1 multiplex using pre-designed 3-8 decoders.
- Then we will have verify all of their functionality by running a simulation using XSim which is already available in the Vivado software.

VHDL Codes (2-4 Decoder)

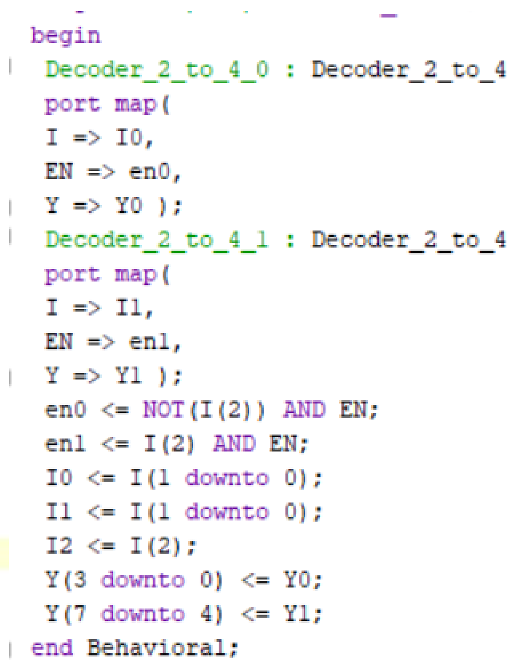


```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- 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;
```

VHDL Codes (3-8 Decoder) (TB)

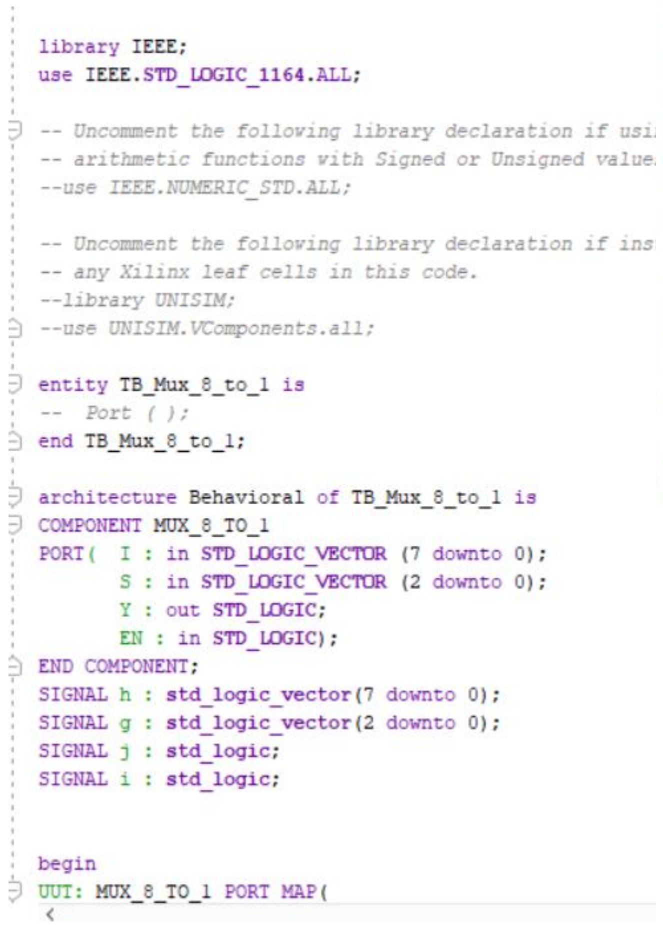


```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- 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
```

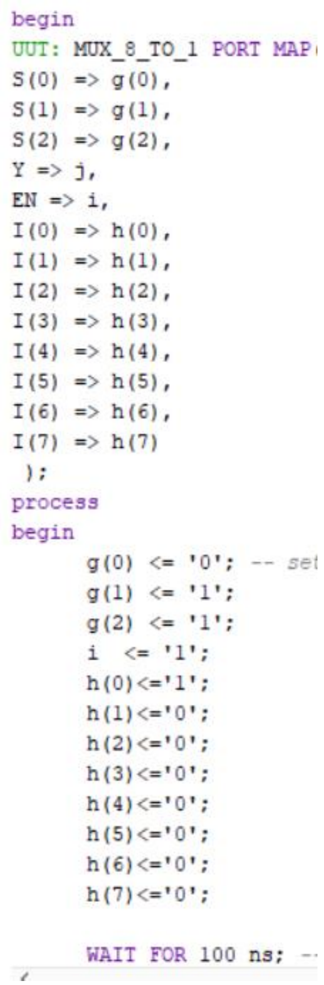


```vhdl
begin
 Decoder_2_to_4_0 : Decoder_2_to_4
 port map(
 I => I0,
 EN => en0,
 Y => Y0 );
 Decoder_2_to_4_1 : Decoder_2_to_4
 port map(
 I => I1,
 EN => en1,
 Y => Y1 );
 en0 <= NOT(I(2)) AND EN;
 en1 <= I(2) AND EN;
 I0 <= I(1 downto 0);
 I1 <= I(1 downto 0);
 I2 <= I(2);
 Y(3 downto 0) <= Y0;
 Y(7 downto 4) <= Y1;
end Behavioral;
```

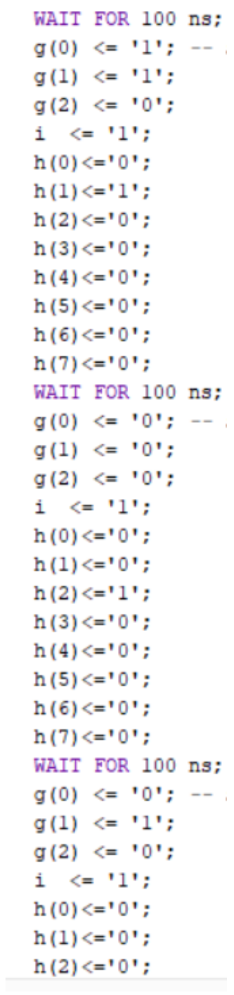
VHDL Codes (8-1 Multiplex) (TB)



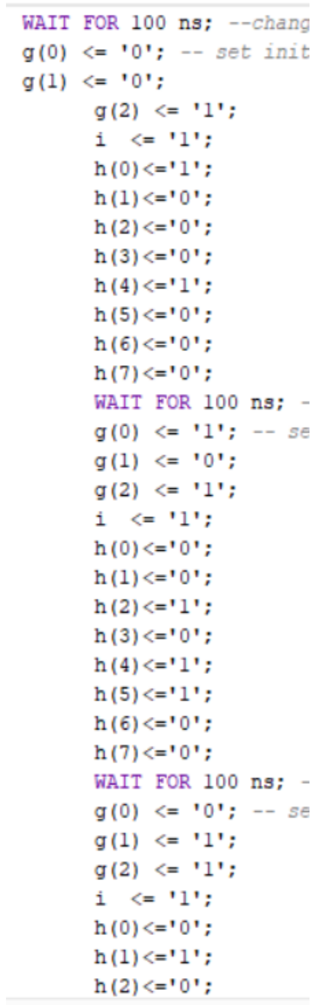
```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

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

-- Uncomment the following library declaration if ins
-- 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(  I : in STD_LOGIC_VECTOR (7 downto 0);
       S : in STD_LOGIC_VECTOR (2 downto 0);
       Y : out STD_LOGIC;
       EN : in STD_LOGIC);
END COMPONENT;
SIGNAL h : std_logic_vector(7 downto 0);
SIGNAL g : std_logic_vector(2 downto 0);
SIGNAL j : std_logic;
SIGNAL i : std_logic;


begin
UUT: MUX_8_TO_1 PORT MAP(
```



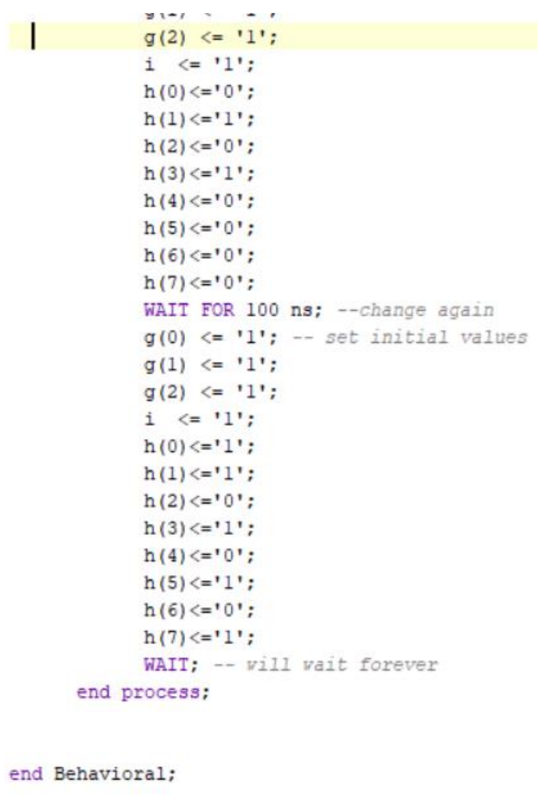
```vhdl
begin
UUT: MUX_8_TO_1 PORT MAP(
S(0) => g(0),
S(1) => g(1),
S(2) => g(2),
Y => j,
EN => i,
I(0) => h(0),
I(1) => h(1),
I(2) => h(2),
I(3) => h(3),
I(4) => h(4),
I(5) => h(5),
I(6) => h(6),
I(7) => h(7)
 );
process
begin
        g(0) <= '0'; -- set
        g(1) <= '1';
        g(2) <= '1';
        i  <= '1';
        h(0)<='1';
        h(1)<='0';
        h(2)<='0';
        h(3)<='0';
        h(4)<='0';
        h(5)<='0';
        h(6)<='0';
        h(7)<='0';

        WAIT FOR 100 ns; --
```



```vhdl
WAIT FOR 100 ns;
g(0) <= '1'; --
g(1) <= '1';
g(2) <= '0';
i  <= '1';
h(0)<='0';
h(1)<='1';
h(2)<='0';
h(3)<='0';
h(4)<='0';
h(5)<='0';
h(6)<='0';
h(7)<='0';
WAIT FOR 100 ns;
g(0) <= '0'; --
g(1) <= '0';
g(2) <= '0';
i  <= '1';
h(0)<='0';
h(1)<='0';
h(2)<='1';
h(3)<='0';
h(4)<='0';
h(5)<='0';
h(6)<='0';
h(7)<='0';
WAIT FOR 100 ns;
g(0) <= '0'; --
g(1) <= '1';
g(2) <= '0';
i  <= '1';
h(0)<='0';
h(1)<='0';
h(2)<='0';
```

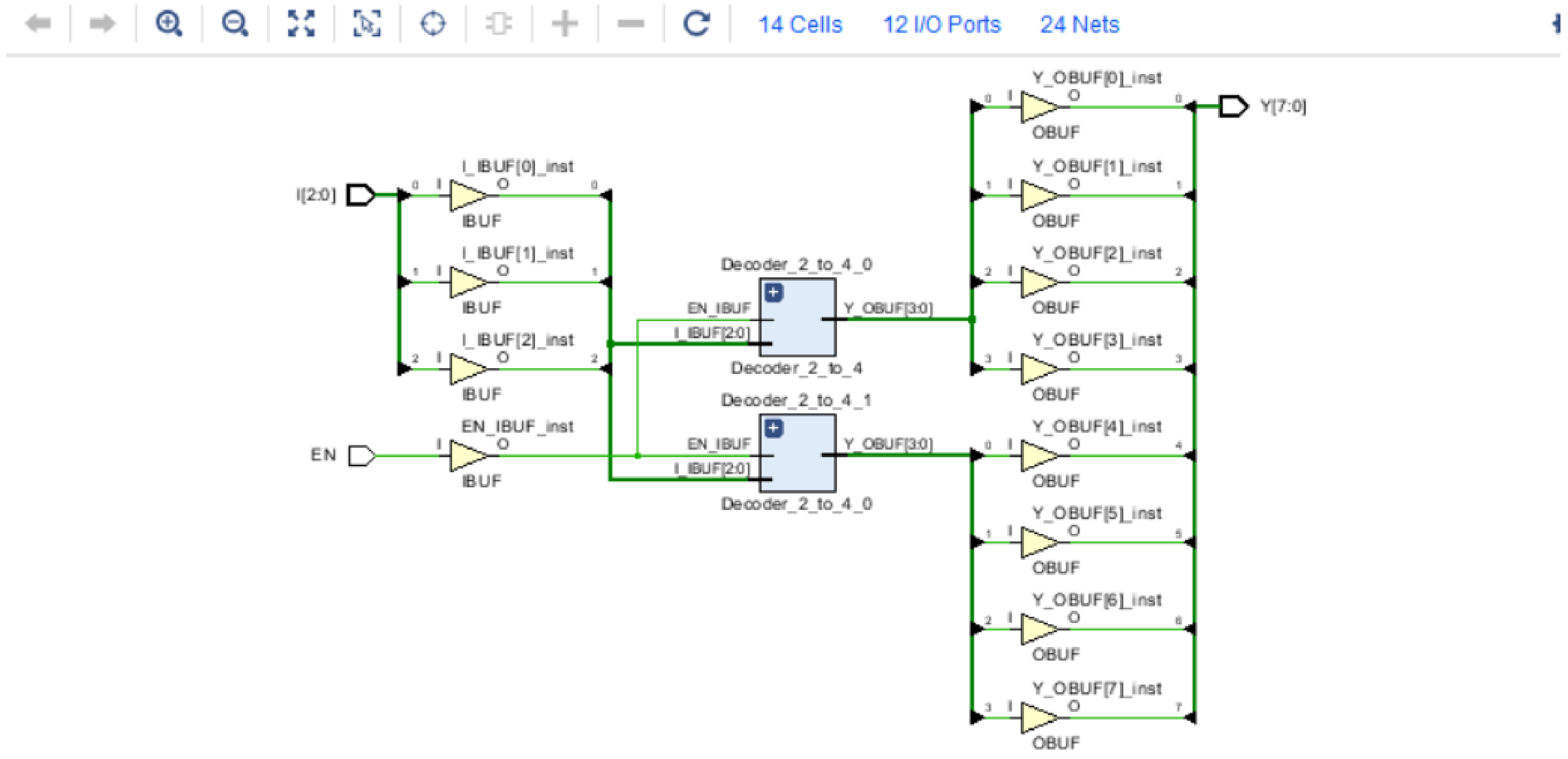


```vhdl
WAIT FOR 100 ns; --chang
g(0) <= '0'; -- set init
g(1) <= '0';
        g(2) <= '1';
        i  <= '1';
        h(0)<='1';
        h(1)<='0';
        h(2)<='0';
        h(3)<='0';
        h(4)<='1';
        h(5)<='0';
        h(6)<='0';
        h(7)<='0';
        WAIT FOR 100 ns; -
        g(0) <= '1'; -- se
        g(1) <= '0';
        g(2) <= '1';
        i  <= '1';
        h(0)<='0';
        h(1)<='0';
        h(2)<='1';
        h(3)<='0';
        h(4)<='1';
        h(5)<='1';
        h(6)<='0';
        h(7)<='0';
        WAIT FOR 100 ns; -
        g(0) <= '0'; -- se
        g(1) <= '1';
        g(2) <= '1';
        i  <= '1';
        h(0)<='0';
        h(1)<='1';
        h(2)<='0';
```



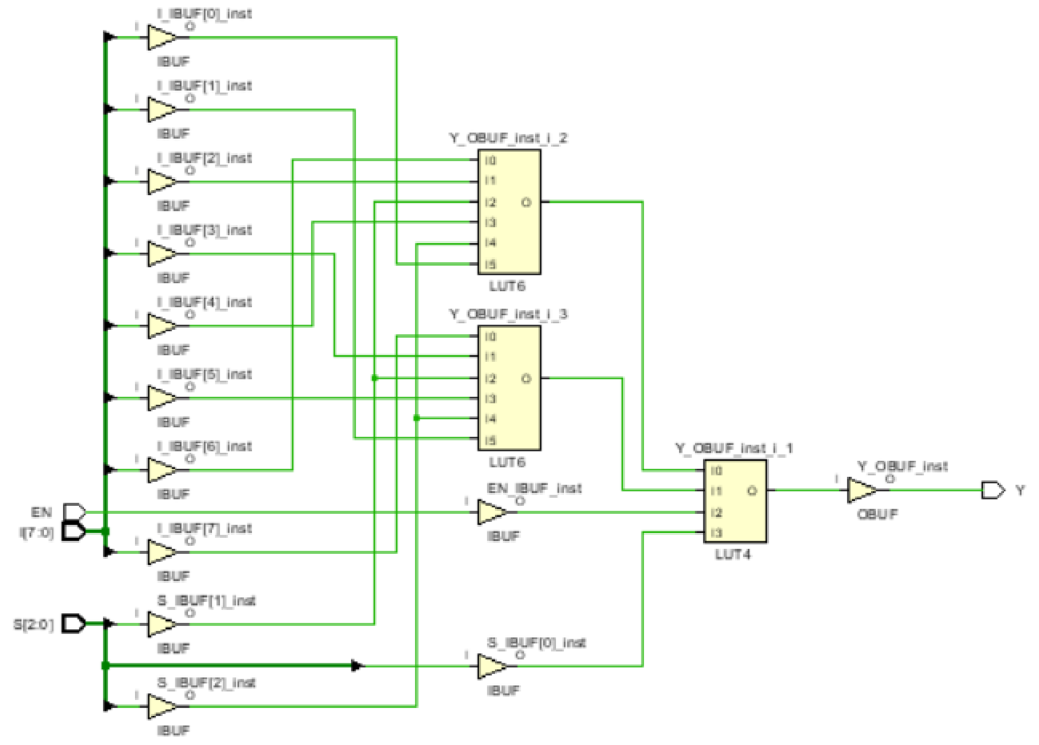
```vhdl
        g(2) <= '1';
        i  <= '1';
        h(0)<='0';
        h(1)<='1';
        h(2)<='0';
        h(3)<='1';
        h(4)<='0';
        h(5)<='0';
        h(6)<='0';
        h(7)<='0';
        WAIT FOR 100 ns; --change again
        g(0) <= '1'; -- set initial values
        g(1) <= '1';
        g(2) <= '1';
        i  <= '1';
        h(0)<='1';
        h(1)<='1';
        h(2)<='0';
        h(3)<='1';
        h(4)<='0';
        h(5)<='1';
        h(6)<='0';
        h(7)<='1';
        WAIT; -- will wait forever
    end process;


end Behavioral;
```

SCHEMATICS (2-4 Decoder)

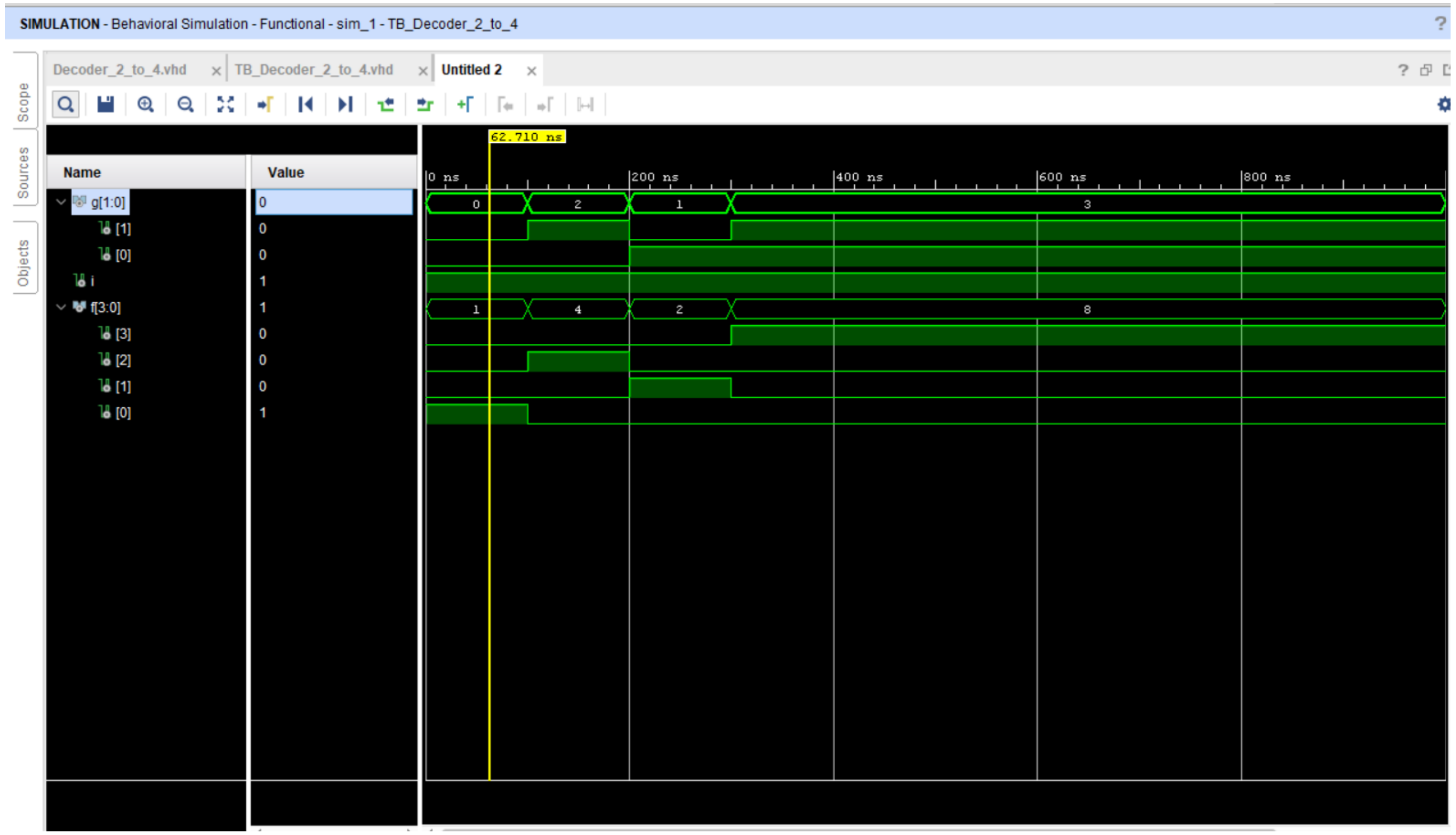


SCHEMATICS (3-8 Decoder)

SCHEMATICS (8-1 MUX)

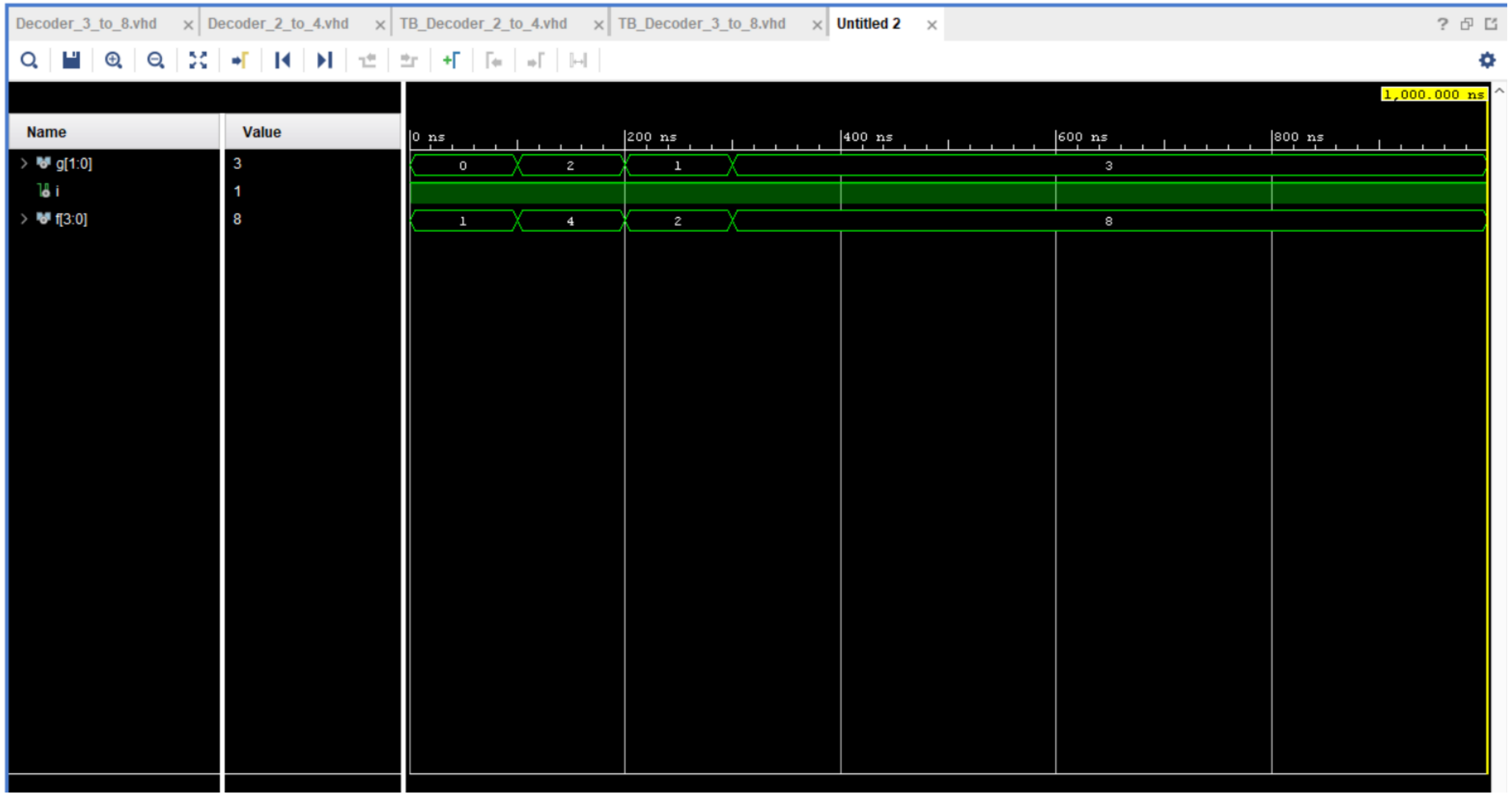


TIMING DIAGRAMS (2-4 Decoder)



Name	Value
g[1:0]	0
[1]	0
[0]	0
i	1
f[3:0]	1
[3]	0
[2]	0
[1]	0
[0]	1

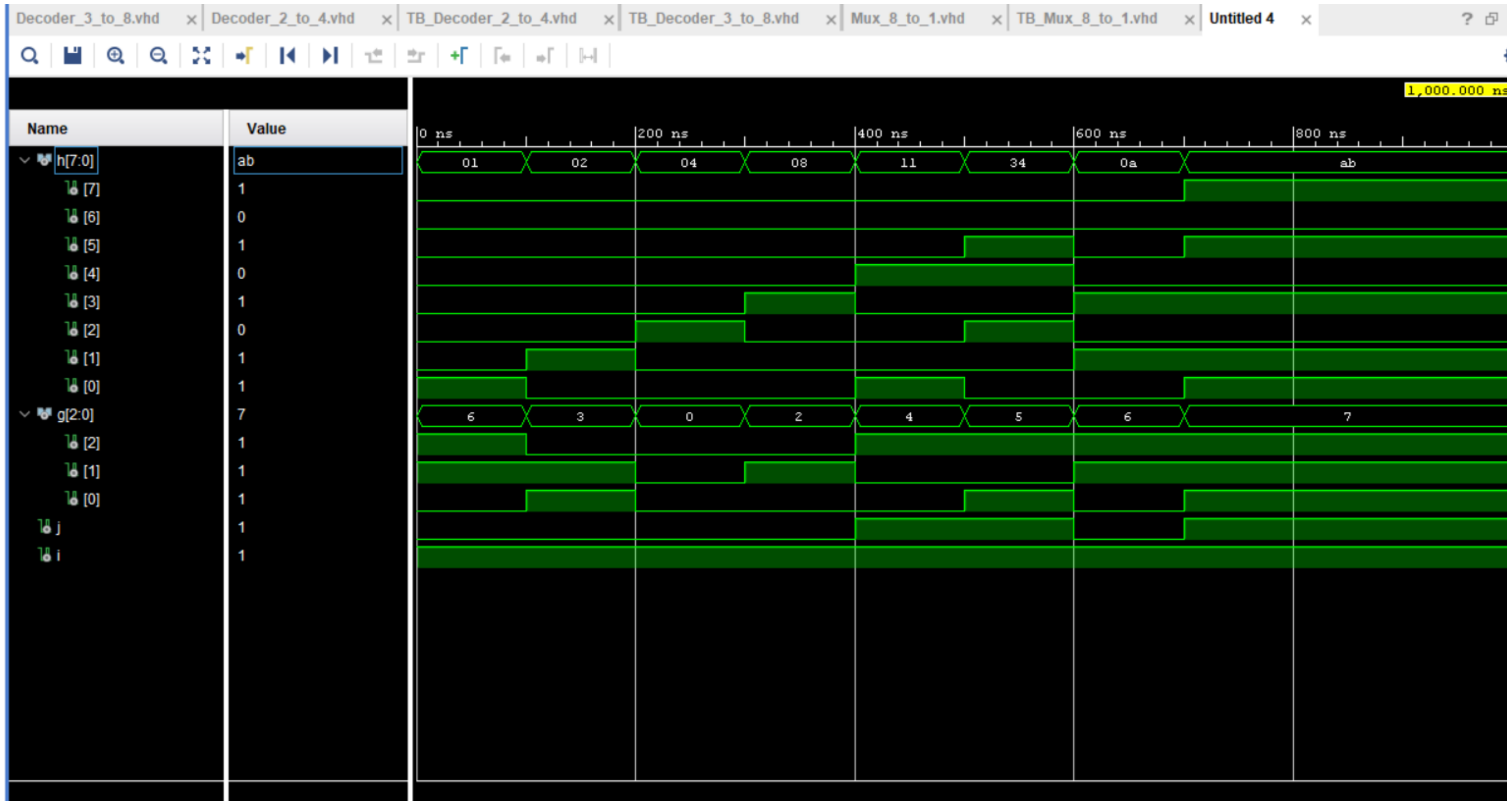
TIMING DIAGRAMS (3-8 Decoder)



Name	Value
g[1:0]	3
i	1
f[3:0]	8

```
g[1:0]:  0 | 2 | 1 | 3
f[3:0]:  1 | 4 | 2 | 8
```

TIMING DIAGRAMS (MUX)



Name	Value
h[7:0]	ab
[7]	1
[6]	0
[5]	1
[4]	0
[3]	1
[2]	0
[1]	1
[0]	1
g[2:0]	7
[2]	1
[1]	1
[0]	1
j	1
i	1

```
h[7:0]:  01  02  04  08  11  34  0a  ab
g[2:0]:  6   3   0   2   4   5   6   7
```

CONCLUSION

From 2-4 decoders, we can built 3-8 decoder hierarchically and from 3-8 decoders, it is easy to build a 8-1 multiplex using the same sort of design.