INSTITUTE OF ENGINEERING AND TECHNOLOGY, INDORE

CIRCUIT DESIGN USING HDL (EIR7C4)

(Lab Assignment)

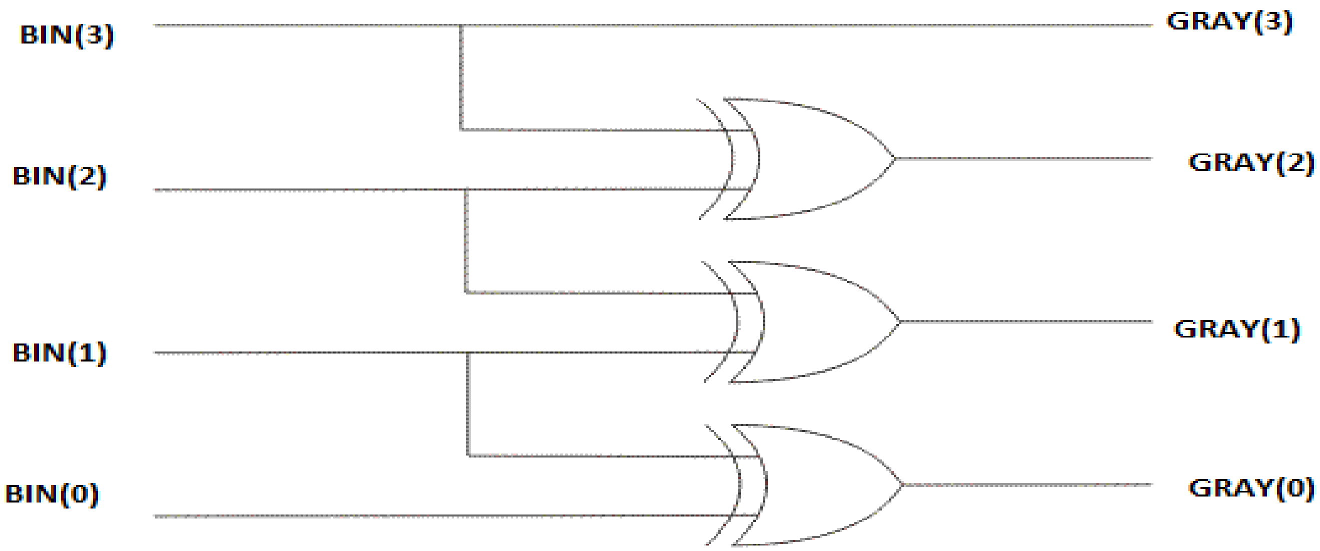
Session (2020-21)

Submitted to:
Dr. Vaibhav Neema

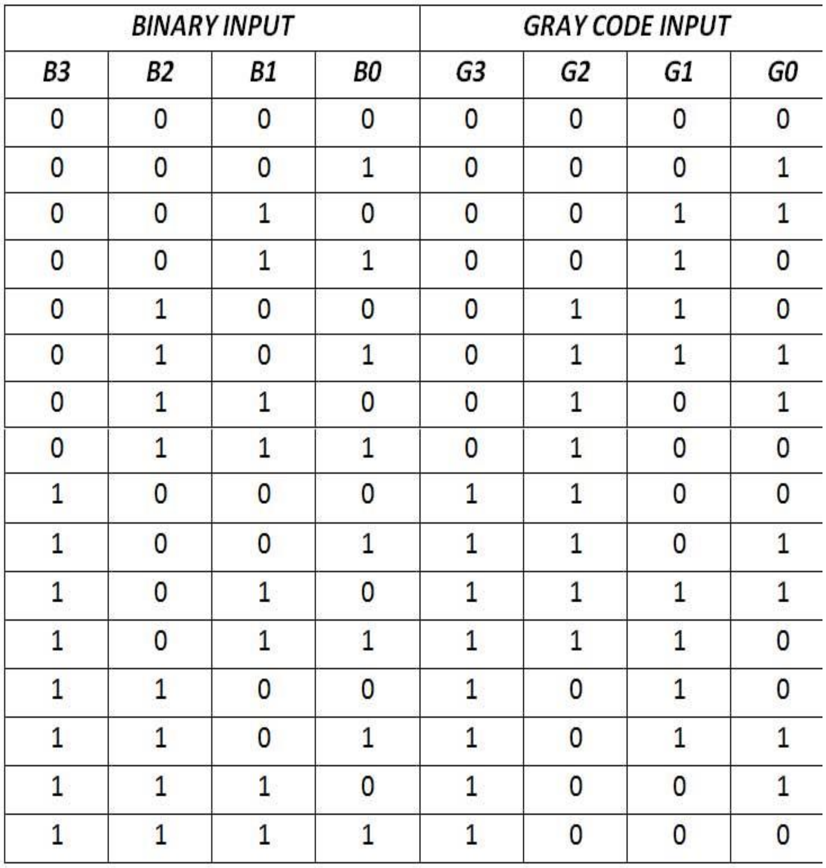
Submitted by
Ayur Sarawgi
E&I (7th Sem)
Roll no. 17E7013
Enroll no. DE17205

Q. Design of Binary to gray code converter using data flow modelling.
Ans.

Circuit of 4 bit binary to gray code converter



Truth Table of Binary to Gray code



BINARY INPUT				GRAY CODE INPUT			
B3	B2	B1	B0	G3	G2	G1	G0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Logic used -

GRAY (3) = BIN (3)

GRAY (2) = BIN (3) **XOR** BIN (2)

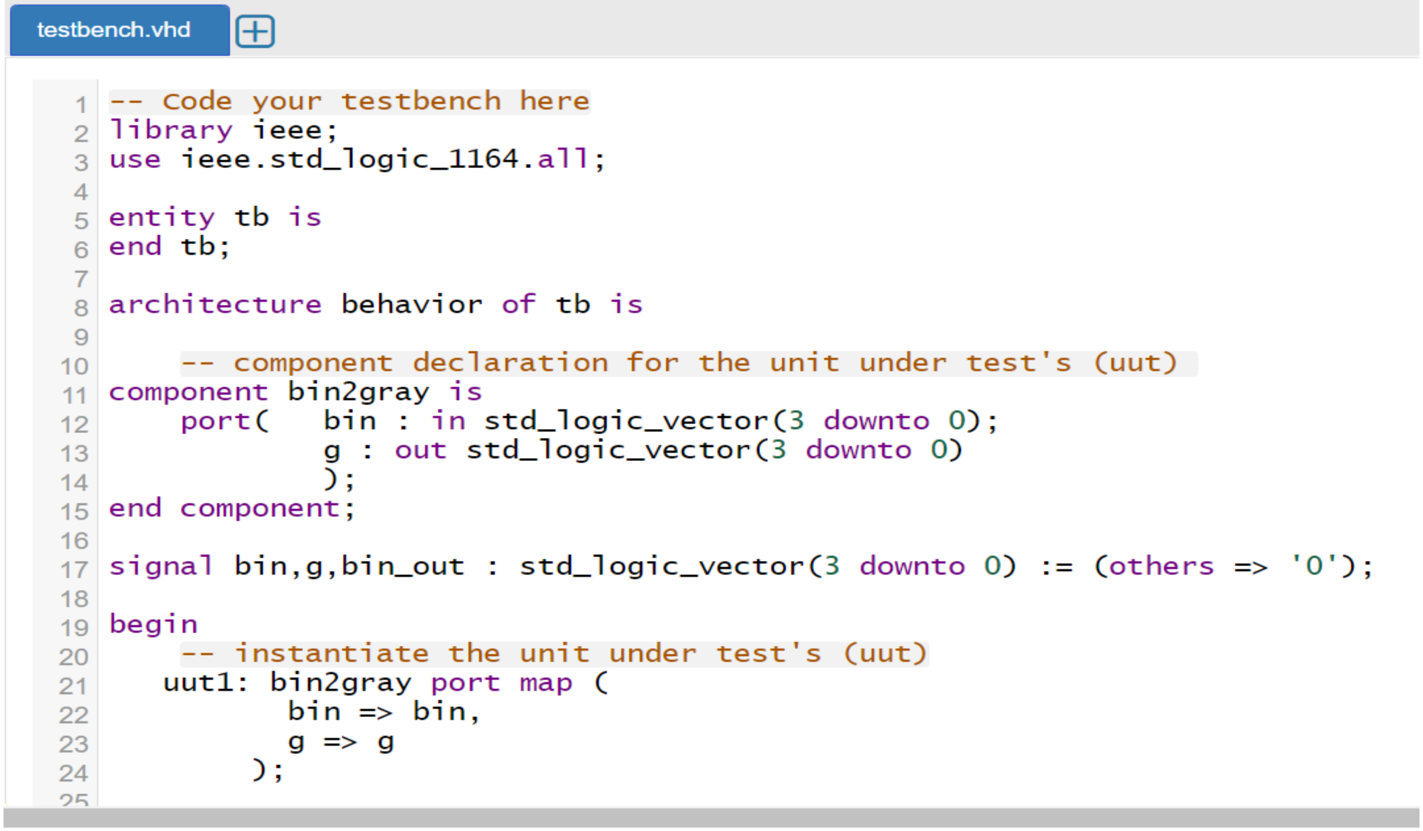
GRAY (1) = BIN (2) **XOR** BIN (1)

GRAY (0) = BIN (1) **XOR** BIN (0)

Component-

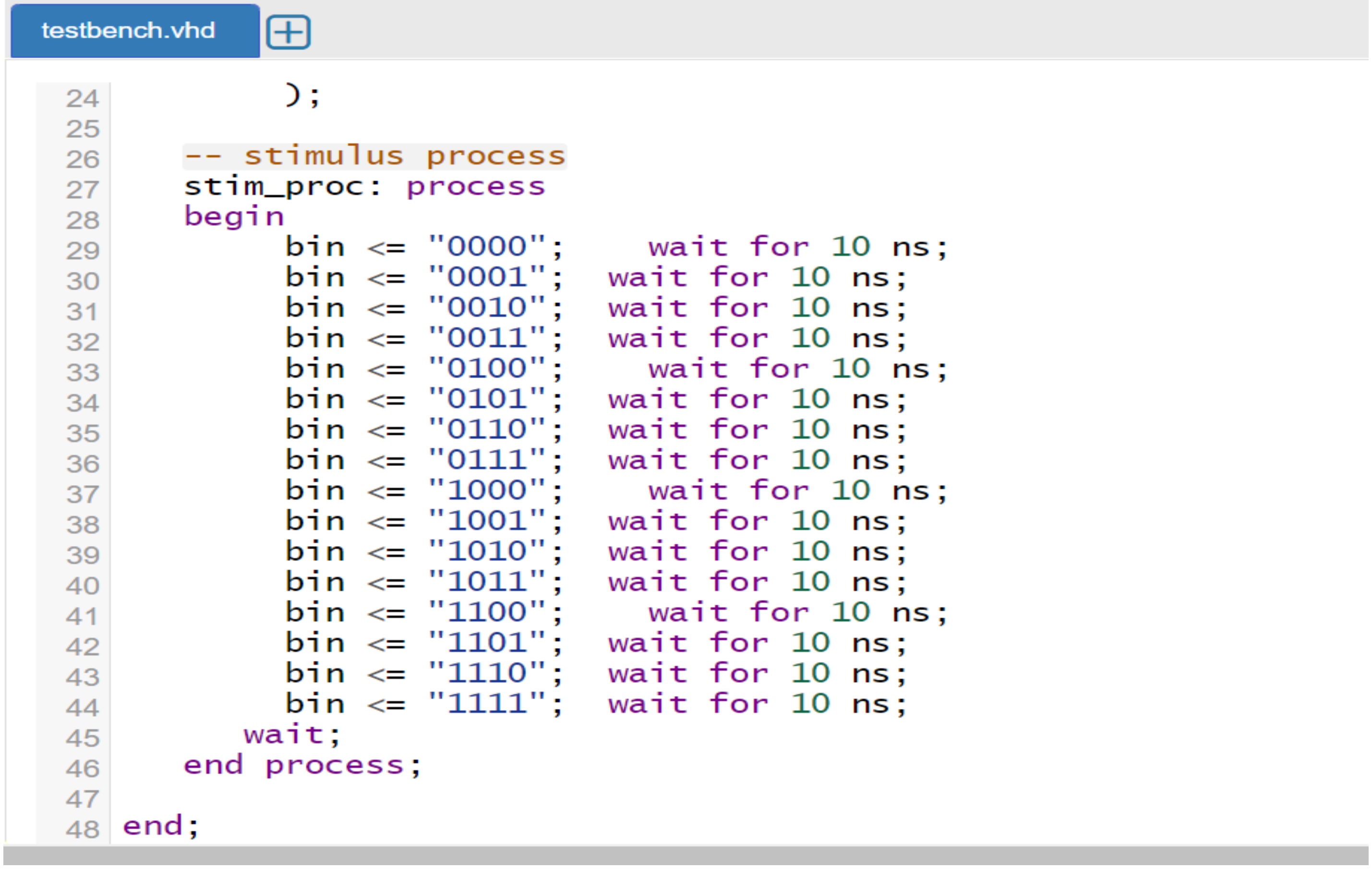
3 XOR gates

Testbench code of 4 bits binary to gray code converter



testbench.vhd

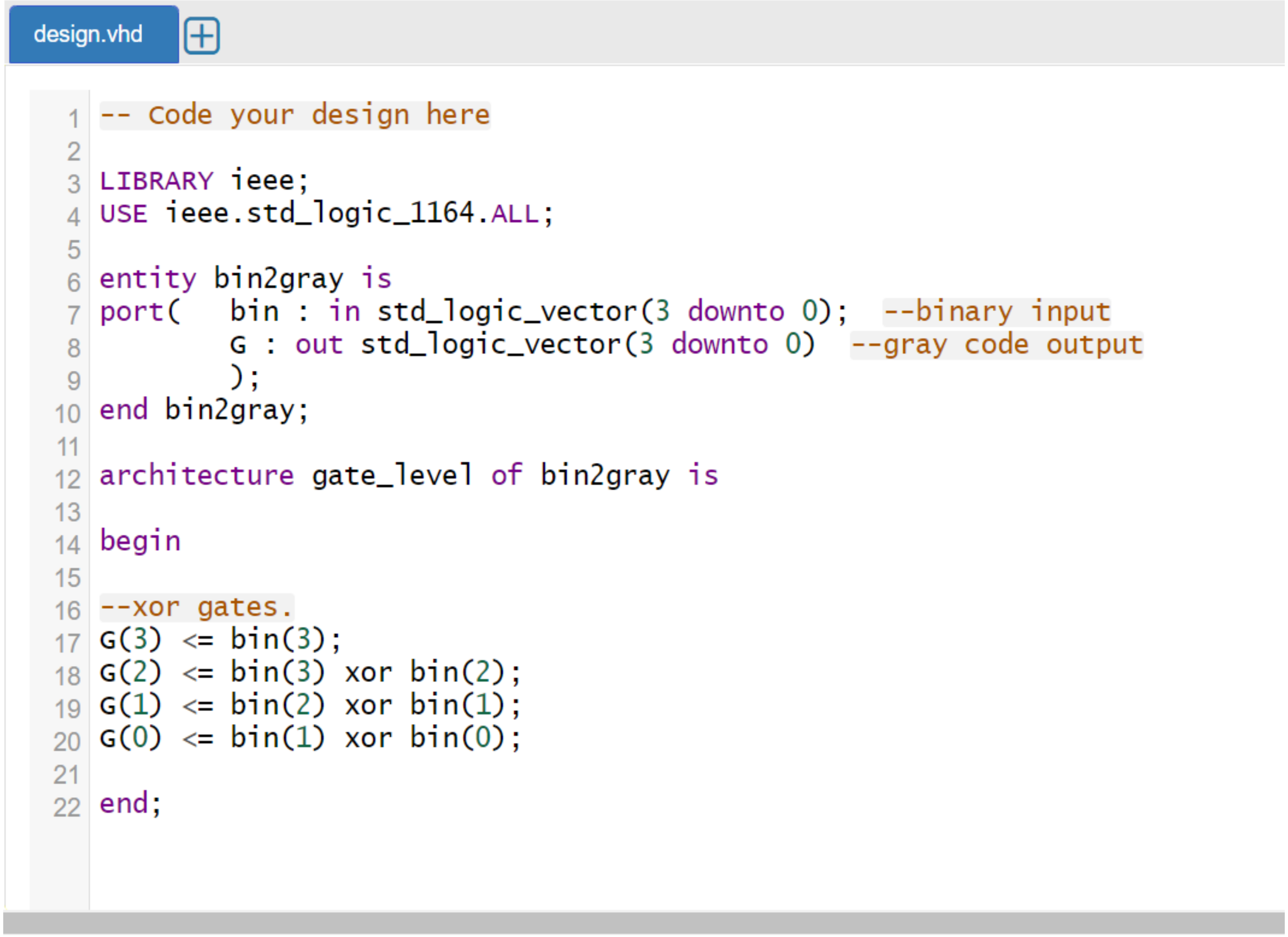
```
-- Code your testbench here
library ieee;
use ieee.std_logic_1164.all;

entity tb is
end tb;

architecture behavior of tb is

    -- component declaration for the unit under test's (uut)
component bin2gray is
    port(   bin : in std_logic_vector(3 downto 0);
            g : out std_logic_vector(3 downto 0)
            );
end component;

signal bin,g,bin_out : std_logic_vector(3 downto 0) := (others => '0');

begin
    -- instantiate the unit under test's (uut)
   uut1: bin2gray port map (
          bin => bin,
          g => g
        );
```



testbench.vhd

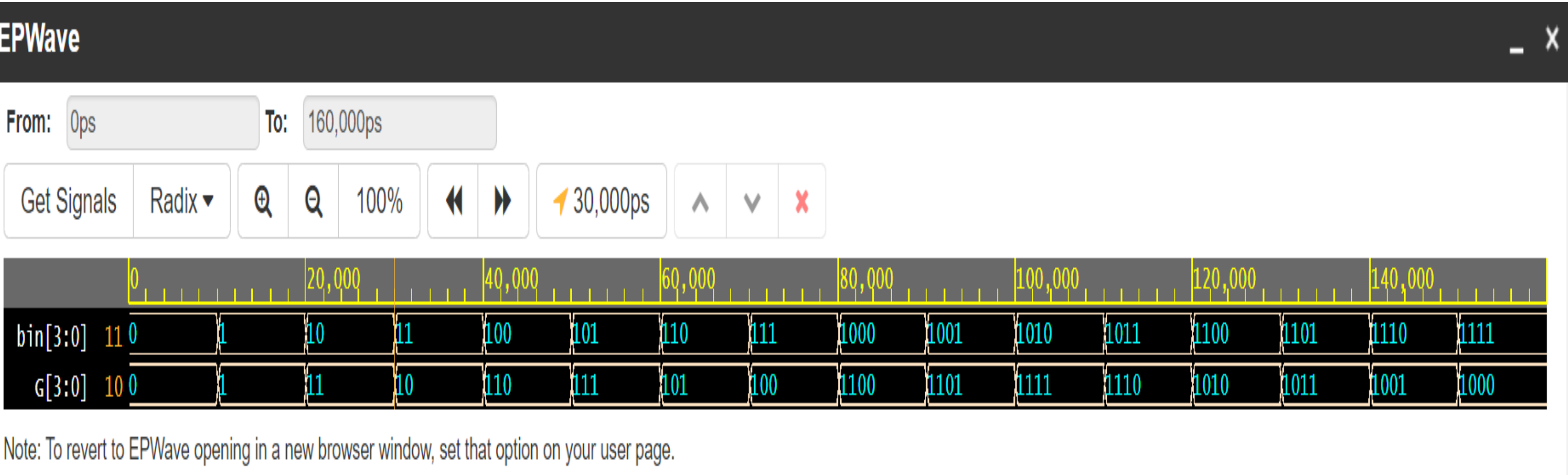
```
        );

   -- stimulus process
   stim_proc: process
   begin
        bin <= "0000";    wait for 10 ns;
        bin <= "0001";  wait for 10 ns;
        bin <= "0010";  wait for 10 ns;
        bin <= "0011";  wait for 10 ns;
        bin <= "0100";    wait for 10 ns;
        bin <= "0101";  wait for 10 ns;
        bin <= "0110";  wait for 10 ns;
        bin <= "0111";  wait for 10 ns;
        bin <= "1000";    wait for 10 ns;
        bin <= "1001";  wait for 10 ns;
        bin <= "1010";  wait for 10 ns;
        bin <= "1011";  wait for 10 ns;
        bin <= "1100";    wait for 10 ns;
        bin <= "1101";  wait for 10 ns;
        bin <= "1110";  wait for 10 ns;
        bin <= "1111";  wait for 10 ns;
      wait;
   end process;

end;
```

Design code of 4 bits binary to gray code converter



```vhdl
-- Code your design here

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

entity bin2gray is
port(   bin : in std_logic_vector(3 downto 0);   --binary input
        G : out std_logic_vector(3 downto 0)   --gray code output
        );
end bin2gray;

architecture gate_level of bin2gray is

begin

--xor gates.
G(3) <= bin(3);
G(2) <= bin(3) xor bin(2);
G(1) <= bin(2) xor bin(1);
G(0) <= bin(1) xor bin(0);

end;
```

Output of 4 bits binary to gray code converter



```
bin[3:0] 11 0 1 10 11 100 101 110 111 1000 1001 1010 1011 1100 1101 1110 1111
G[3:0]   10 0 1 11 10 110 111 101 100 1100 1101 1111 1110 1010 1011 1001 1000
```

Note: To revert to EPWave opening in a new browser window, set that option on your user page.