

INSTITUTE OF ENGINEERING AND TECHNOLOGY, INDORE



CIRCUIT DESIGN USING HDL (EIR7C4)

(Lab Assignment)

Session (2020-21)

Submitted to:

Dr. Vaibhav Neema

Submitted by

Shrey Gaur

E&I (7th Sem)

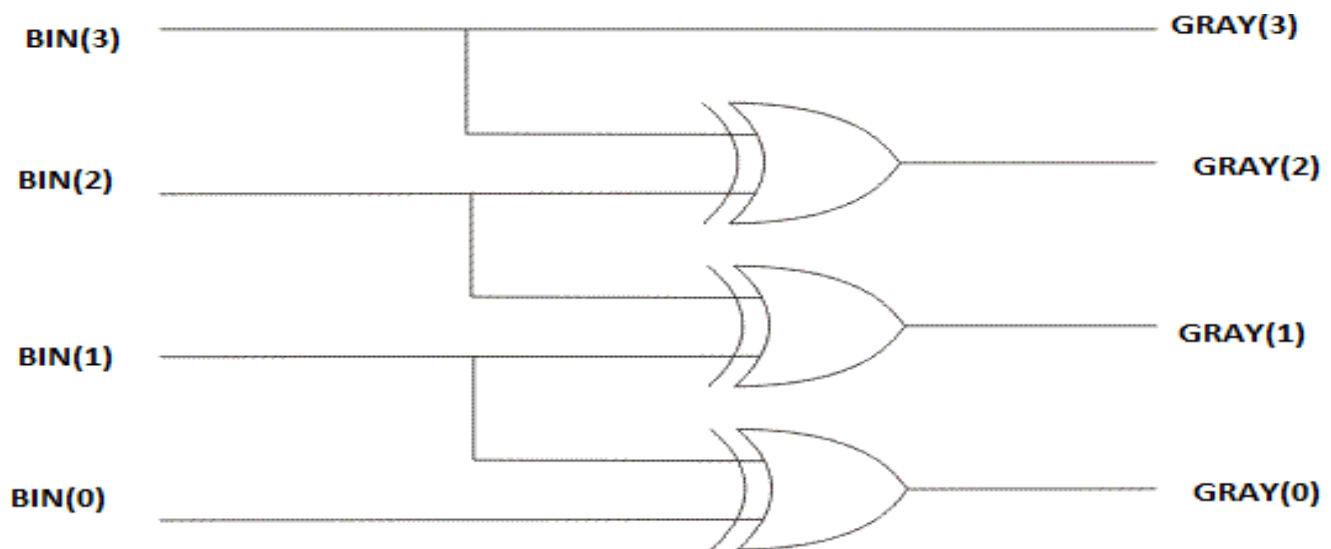
Roll no. 17E7048

Enroll no. DE17239

Q. Design of Binary to gray code converter using structure 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)

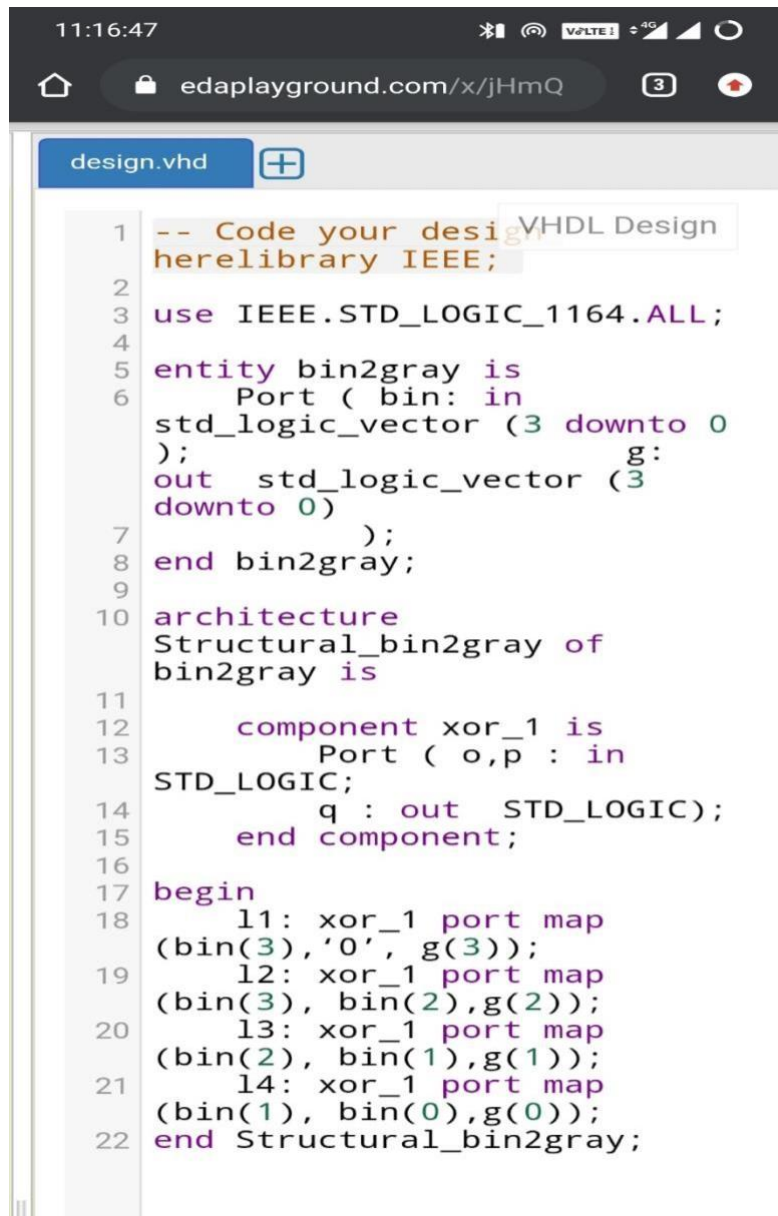
GRAY (0) = BIN (1) **XOR** BIN (0)

Component-

3 XOR gates

Binary to gray code converter in vhdl using structural modelling:

Design code:



The screenshot shows a web browser interface with the address bar displaying 'edaplayground.com/x/jHmQ'. The page title is 'design.vhd'. The code is written in VHDL and implements a binary to gray code converter using structural modeling. The code is as follows:

```
1  -- Code your design here
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4
5  entity bin2gray is
6      Port ( bin: in
7             std_logic_vector (3 downto 0)
8             );
9      out std_logic_vector (3
10         downto 0)
11         );
12 end bin2gray;
13
14 architecture
15     Structural_bin2gray of
16     bin2gray is
17
18         component xor_1 is
19             Port ( o,p : in
20                   STD_LOGIC;
21                   q : out  STD_LOGIC);
22         end component;
23
24 begin
25     l1: xor_1 port map
26         (bin(3), '0', g(3));
27     l2: xor_1 port map
28         (bin(3), bin(2), g(2));
29     l3: xor_1 port map
30         (bin(2), bin(1), g(1));
31     l4: xor_1 port map
32         (bin(1), bin(0), g(0));
33 end Structural_bin2gray;
```

Testbench code:

```
testbench.vhd
1  -- Code your testbench
2  library ieee;
3  use ieee.std_logic_1164.all;
4
5  entity tb is
6  end tb;
7
8  architecture behavior of tb
9  is
10     -- component declaration
11     for the unit under test's
12     (uut)
13     component bin2gray is
14     port( bin : in
15           std_logic_vector(3 downto 0));
16           g : out
17           std_logic_vector(3 downto 0)
18           );
19     end component;
20
21     signal bin,g;
22     std_logic_vector(3 downto 0)
23     := (others=>'0');
24
25 begin
26     -- instantiate the unit
27     under test's (uut)
28     uut1: bin2gray port map (
29         bin => bin,
30         g => g
31     );
32
33     -- stimulus process
34     stim_proc: process
35     begin
36         bin <= "0000";
37         wait for 10 ns;
38         bin <= "0001"; wait
39         for 10 ns;
40         bin <= "0010"; wait
41         for 10 ns;
42         bin <= "0011"; wait
43         for 10 ns;
44         bin <= "0100";
45         wait for 10 ns;
46         bin <= "0101"; wait
47         for 10 ns;
48         bin <= "0110"; wait
49         for 10 ns;
50         bin <= "0111"; wait
51         for 10 ns;
52         bin <= "1000";
53         wait for 10 ns;
54         bin <= "1001"; wait
55         for 10 ns;
56         bin <= "1010"; wait
57         for 10 ns;
58         bin <= "1011"; wait
59         for 10 ns;
60         bin <= "1100";
61         wait for 10 ns;
62         bin <= "1101"; wait
63         for 10 ns;
64         bin <= "1110"; wait
65         for 10 ns;
66         bin <= "1111"; wait
67         for 10 ns;
68         wait;
69     end process;
70
71 end;
```

EP waveform:

