INSTITUTE OF ENGINEERING AND TECHNOLOGY, INDORE

EIR7E1: Circuit Design Using VHDL

“Design of Binary to Gray code converter”

Submitted To:

Dr. Vaibhav Neema

Submitted By:

Vishvas Choudhary

17E7059

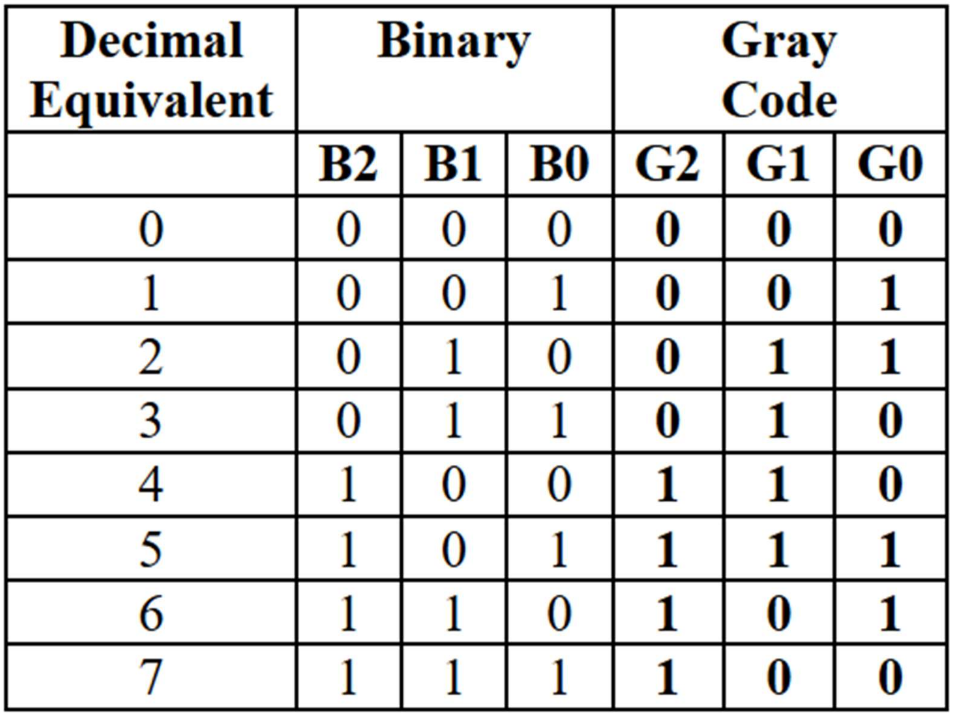
Code Converter - Binary to Gray

Gray Code system is a binary number system in which every successive pair of numbers differs in only one bit. It is used in applications in which the normal sequence of binary numbers generated by the hardware may produce an error or ambiguity during the transition from one number to the next.

For example, the representation of the decimal value "1" in binary would normally be "001" and "2" would be "010". In Gray code, these values are represented as "001" and "011". That way, incrementing a value from 1 to 2 requires only one bit to change, instead of two.

Converting Binary to Gray Code-

Let B_0, B_1, B_2, B_3 be the bits representing the binary numbers, where B_0 is the LSB and B_3 is the MSB, and
Let G_0, G_1, G_2, G_3 be the bits representing the gray code of the binary numbers, where G_0 is the LSB and G_3 is the MSB.



Decimal Equivalent	Binary			Gray Code		
	B2	B1	B0	G2	G1	G0
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	1
3	0	1	1	0	1	0
4	1	0	0	1	1	0
5	1	0	1	1	1	1
6	1	1	0	1	0	1
7	1	1	1	1	0	0

Boolean Equation for the outputs G_0, G_1, G_2 and G_3 can be given as:

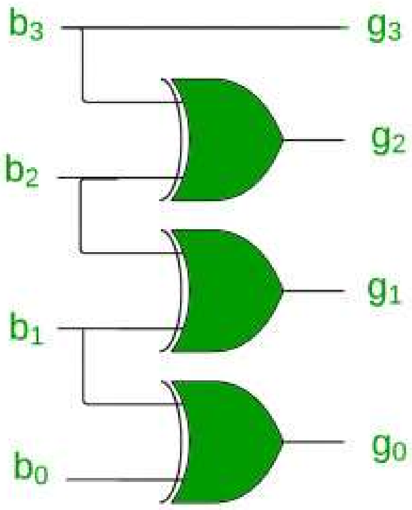
$G_0 = B_0 \oplus B_1$

$G_1 = B_1 \oplus B_2$

$G_2 = B_2 \oplus B_3$

$G_3 = B_3$

These Equations can be implemented using 3-input XOR gates



VHDL Code for converting Binary to Gray Code

DESIGN CODE

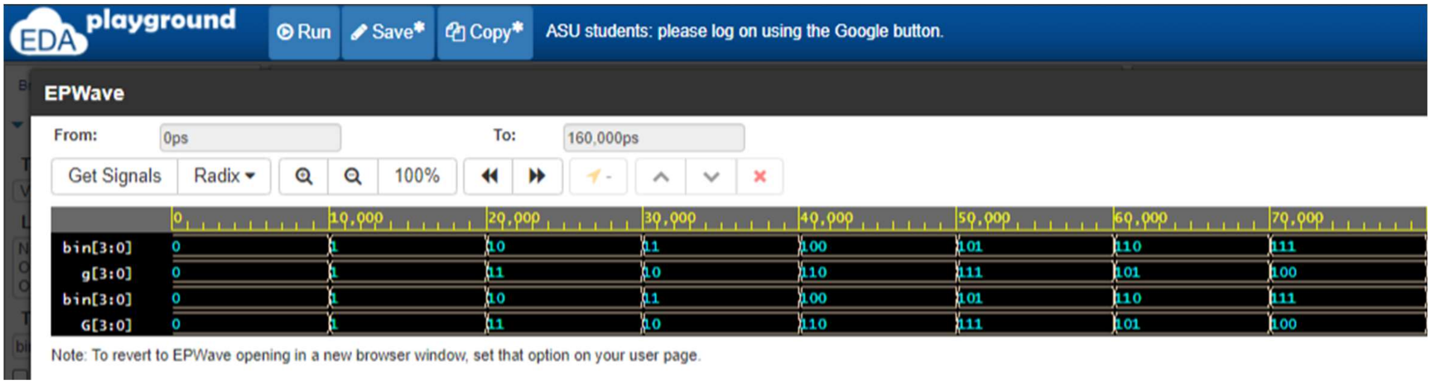
```
library ieee;

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.

        process(bin)

                begin

                        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 process;

end;
```

TESTBENCH CODE

```
library ieee;

use ieee.std_logic_1164.ALL;

entity bin2gray_tb is

end bin2gray_tb;

architecture behavioral of bin2gray_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: std_logic_vector(3 downto 0) := (others => '0');

begin

   -- instantiate the unit under test's (uut)

  uut1: bin2gray port map (

       bin => bin,

       g => g

     );

  -- 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;
```

SIMULATION WAVEFORM



ASU students: please log on using the Google button.

From: 0ps

To: 160,000ps

```
           0      10,000  20,000  30,000  40,000  50,000  60,000  70,000
bin[3:0]   0      1       10      11      100     101     110     111
  g[3:0]   0      1       11      10      110     111     101     100
bin[3:0]   0      1       10      11      100     101     110     111
  G[3:0]   0      1       11      10      110     111     101     100
```

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

Website to visit the code:-
https://www.edaplayground.com/x/qHCg