EXPERIMENT No-7

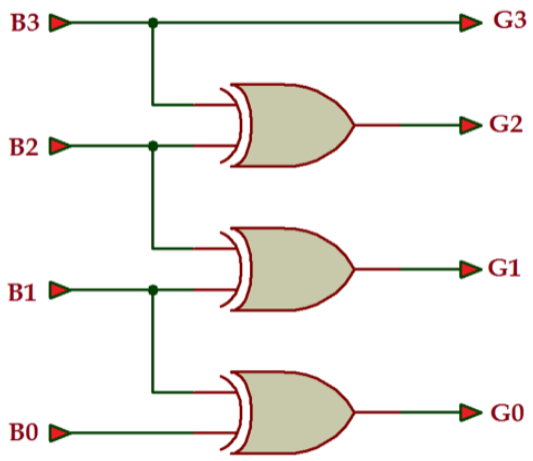
Aim: To Design 4-bit binary-to-gray and gray-to-binary code converters using VHDL

Description:

1. **The 4-bit, binary-to-gray code converter**

Using Exclusive-Or (⊕) operation: These are following steps for n-bit binary numbers -

- The most significant bit (MSB) of the Gray code is always equal to the MSB of the given Binary code.
- Other bits of the output Gray code can be obtained by XORing binary code bit at the index and previous index.



Truth table

B3	B2	B1	B0	Binary code	G3	G2	G1	G0	Gray code
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	1	1
0	0	1	0	2	0	0	1	1	3
0	0	1	1	3	0	0	1	0	2
0	1	0	0	4	0	1	1	0	6
0	1	0	1	5	0	1	1	1	7
0	1	1	0	6	0	1	0	1	5
0	1	1	1	7	0	1	0	0	4
1	0	0	0	8	1	1	0	0	12
1	0	0	1	9	1	1	0	1	13
1	0	1	0	10	1	1	1	1	15
1	0	1	1	11	1	1	1	0	14
1	1	0	0	12	1	0	1	0	10
1	1	0	1	13	1	0	1	1	11
1	1	1	0	14	1	0	0	1	9
1	1	1	1	15	1	0	0	0	8

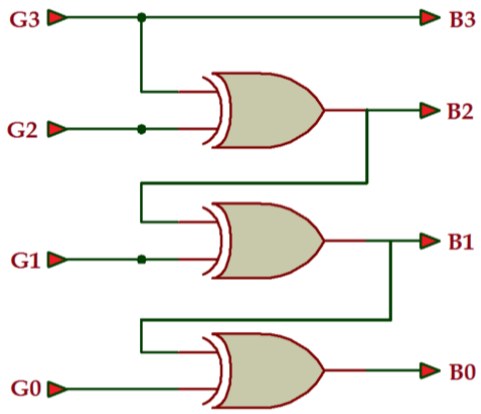
VHDL program:

```
library ieee;
use ieee.std_logic_1164.all;
entity b2g_code is
port (b : in std_logic_vector(3 downto 0);
     g : out std_logic_vector(3 downto 0));
end b2g_code;
architecture b2g_arch of b2g_code is
 begin
g(3) <= b(3);
g(2) <= b(3) xor b(2);
g(1) <= b(2) xor b(1);
g(0) <= b(1) xor b(0);
  end b2g_arch;
```

Output:

2. The gray-to-binary code converter circuit

- The Most Significant Bit (MSB) of the binary code is always equal to the MSB of the given gray code.
- Other bits of the output binary code can be obtained by checking gray code bit at that index. If current gray code bit is 0, then copy previous binary code bit, else copy invert of previous binary code bit.



VHDL program:

```
library ieee;
use ieee.std_logic_1164.all;
entity g2b_code is
port (g : in std_logic_vector(3 downto 0);
      b : out std_logic_vector(3 downto 0));
end g2b_code;
architecture g2b_arch of g2b_code is
  begin
    b(3) <= g(3);
    b(2) <= g(3) xor g(2);
    b(1) <= g(3) xor g(2) xor g(1);
    b(0) <= g(3) xor g(2) xor g(1) xor g(0);
end g2b_arch;
```

Output: