VLSI Design – EECE3051

TOPIC – Binary to grey and Grey to Binary

Aim – To design, simulate, and implement the binary to grey and grey to binary using FPGA

Group members:

Bhargavi N – BU21EECE0100501

Greeshma CK – BU1EECE0100523

Under the guidance of :

M ARUN KUMAR sir

Associate professor, ECE department GITAM University Bengaluru

Introduction-

Binary code is a fundamental numerical system used in digital electronics and computing. It represents numbers using combinations of only two symbols: 0 and 1. For example, the decimal number 5 is represented as "101" in binary.

Gray code, on the other hand, is a binary numeral system where two successive values differ in only one bit. This property makes it useful in applications where reducing errors during transitions is crucial, such as in rotary encoders and Analog-to-digital converters.

1] Binary to Grey:

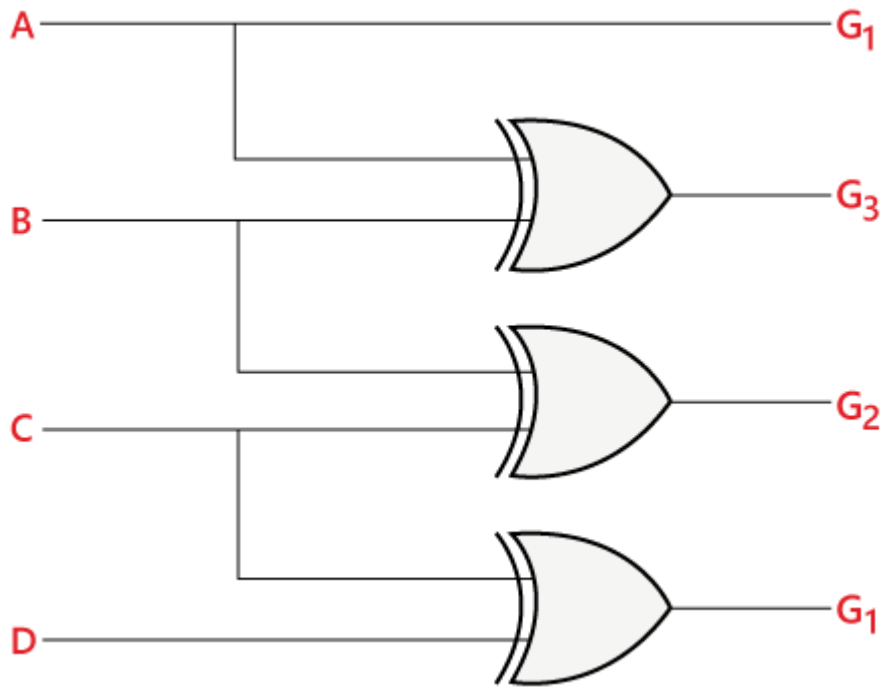
Procedure-

- Start from the most significant bit (MSB) of the binary number.
- The MSB of the Gray code is the same as the MSB of the binary number.
- For each bit in the binary number from left to right, XOR it with the previous bit to get the corresponding Gray code bit.

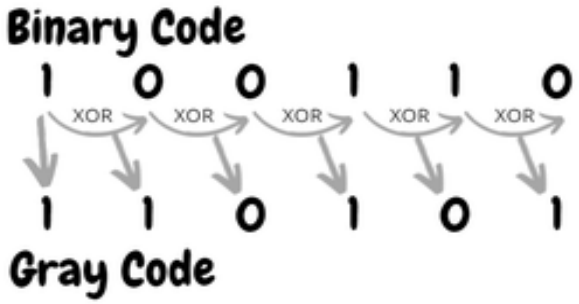
Truth table-

Decimal Number	4-bit Binary Code	4-bit Gray Code
	ABCD	$G_1G_2G_3G_4$
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Circuit diagram-



Example for binary to grey conversion-



Code [gate level modeling]-

```
module binary_to_gray(
    input b1,
    input b2,
```

```
    input b3,
    input b4,
    output g1,
    output g2,
    output g3,
    output g4
    );
buf(g1,b1);
xor (g2,b1,b2),(g3,b2,b3),(g4,b3,b4);
endmodule
```

[case statement]-

```
module b_g( din ,dout );

input [3:0] din ;
wire [3:0] din ;

output [3:0] dout ;
reg [3:0] dout ;

always @ (din) begin
 case (din)
  4'b0000 : dout = 0000;
  4'b0001 : dout = 0001;
  4'b0010 : dout = 0011;
  4'b0011 : dout = 0010;
  4'b0100 : dout = 0110;
  4'b0101 : dout = 0111;
  4'b0110 : dout = 0101;
  4'b0111 : dout = 0100;
  4'b1000 : dout = 1100;
  4'b1001 : dout = 1101;
```

```
 4'b1010 : dout = 1111;
 4'b1011 : dout = 1110;
 4'b1100 : dout = 1010;
 4'b1101 : dout = 1011;
 4'b1110 : dout = 1001;
 default : dout = 1000;
endcase
end

endmodule
```

Output :

Input- 0010

Output- 0011

2] Grey to Binary:

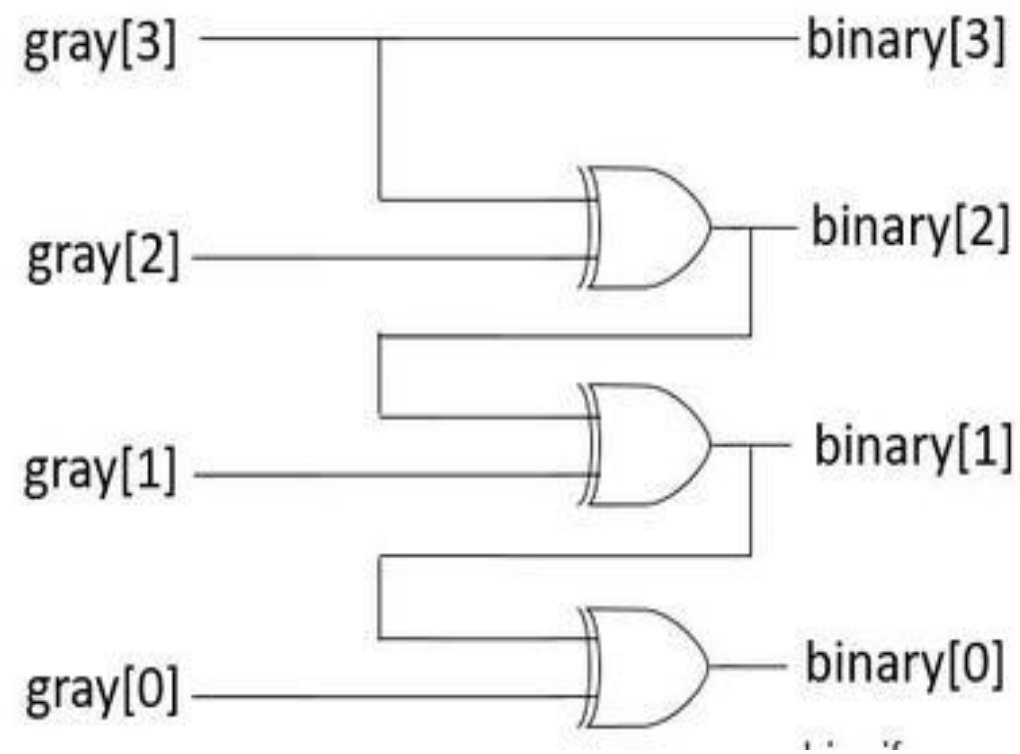
Procedure-

- Start from the MSB of the Gray code.
- The MSB of the binary number is the same as the MSB of the Gray code.
- For each bit in the Gray code from left to right, XOR it with the previous bit of the Gray code to get the corresponding binary bit.

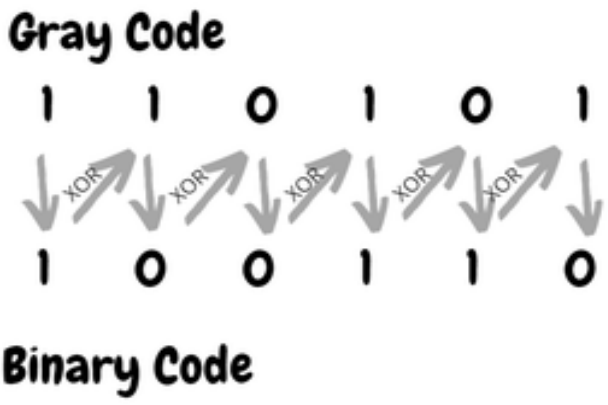
Truth table-

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

Circuit diagram-



Example for binary to grey conversion-



Code [gate level modeling]-

```
module g_b(
   input g0,
   input g1,
   input g2,
   input g3,
   output b0,
   output b1,
   output b2,
   output b3
   );
 buf(b0,g0);
 xor(b1,g0,g1),(b2,g0,g1,g2),(b3,g0,g1,g2,g3);
endmodule
```

[case statement]-

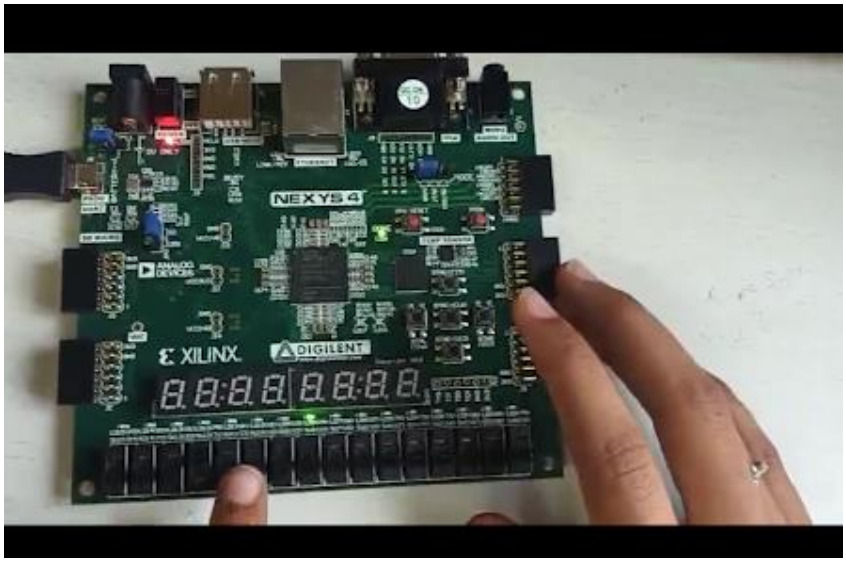
```
module g_b( din ,dout );
```

```
input [3:0] din ;
wire [3:0] din ;

output [3:0] dout ;
reg [3:0] dout ;

always @ (din) begin
 case (din)
  4'b0000 : dout = 0000;
  4'b0001 : dout = 0001;
  4'b0011 : dout = 0010;
  4'b0010 : dout = 0011;
  4'b0110 : dout = 0100;
  4'b0111 : dout = 0101;
  4'b0101 : dout = 0110;
  4'b0100 : dout = 0111;
  4'b1100 : dout = 1000;
  4'b1101 : dout = 1001;
  4'b1111 : dout = 1010;
  4'b1110 : dout = 1011;
  4'b1010 : dout = 1100;
  4'b1011 : dout = 1101;
  4'b1001 : dout = 1110;
  default : dout = 0000;
 endcase
end

endmodule
```

Output-

Input – 0001

Output - 0001



Conclusion:

The conversion between binary and Gray codes involves a simple XOR operation between adjacent bits. Gray codes are useful in applications where reducing errors during transitions is important

Tutorial link for both binary to grey and grey to binary:

https://drive.google.com/file/d/1kaSlBEMSZ7teLUvRl7x_t267jBwp3Ok3/view?usp=sharing

THANK YOU!!!