📘 **Day 12 Topic: Verilog codes for Binary to Gray Code Converter Using Cadence (Xcelium)**

🔹 What is Gray Code?

- Gray code (also called Reflected Binary Code) is a binary number system where two successive numbers differ in only one bit.
- Example (3-bit):
  - Binary: 000, 001, 010, 011, 100, 101, 110, 111
  - Gray: 000, 001, 011, 010, 110, 111, 101, 100

This property helps reduce errors in digital systems like counters, rotary encoders, and communication systems.

🔹 Why do we need Binary to Gray Conversion?

- In binary counting, multiple bits may change at the same time.
  Example: Binary 0111 (7) → 1000 (8) (all 4 bits flip).
- If hardware is slightly slow or noisy, this may cause glitches and wrong readings.
- In Gray code, only 1 bit changes between consecutive numbers, minimizing error.

🔹 Conversion Rule (Binary → Gray)

For an n-bit binary number B[n-1:0], the equivalent Gray code G[n-1:0] is:

- G[n-1] = B[n-1] (MSB remains the same)
- G[i] = B[i+1] XOR B[i] for i = n-2 down to 0

👉 Example (4-bit):
Binary B = b3 b2 b1 b0
Gray G = g3 g2 g1 g0

- g3 = b3
- g2 = b3 XOR b2
- g1 = b2 XOR b1
- g0 = b1 XOR b0

◆ Example Conversion

Binary 1011 (11) ➔ Gray Code:

- g3 = b3 = 1
- g2 = b3 XOR b2 = 1 XOR 0 = 1
- g1 = b2 XOR b1 = 0 XOR 1 = 1
- g0 = b1 XOR b0 = 1 XOR 1 = 0

So, Gray = 1110.

◆ **1.1) Design Code**

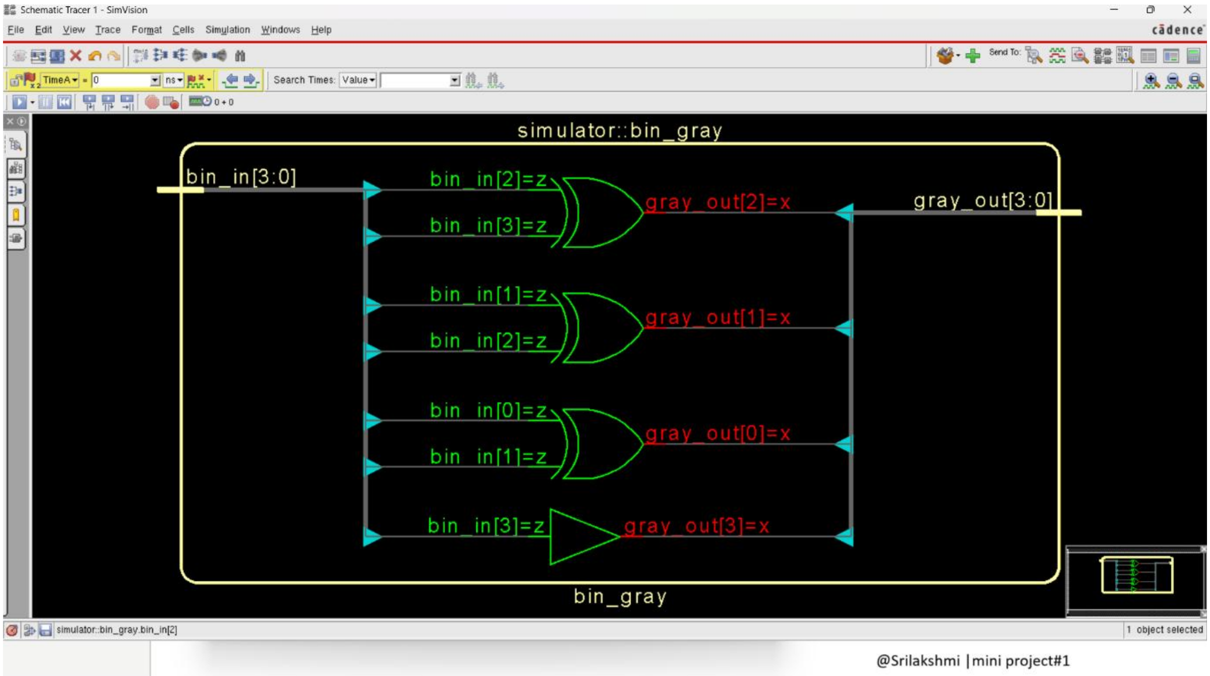
bin_gray.v

```
//DESIGN CODE OF BINARY TO GRAY CODE CONVERSION USING VERILOG

module bin_gray(input [3:0]bin_in, output [3:0]gray_out);
  assign gray_out[3] = bin_in[3];
  assign gray_out[2] = bin_in[2] ^ bin_in[3];
  assign gray_out[1] = bin_in[1] ^ bin_in[2];
  assign gray_out[0] = bin_in[0] ^ bin_in[1];
endmodule
```

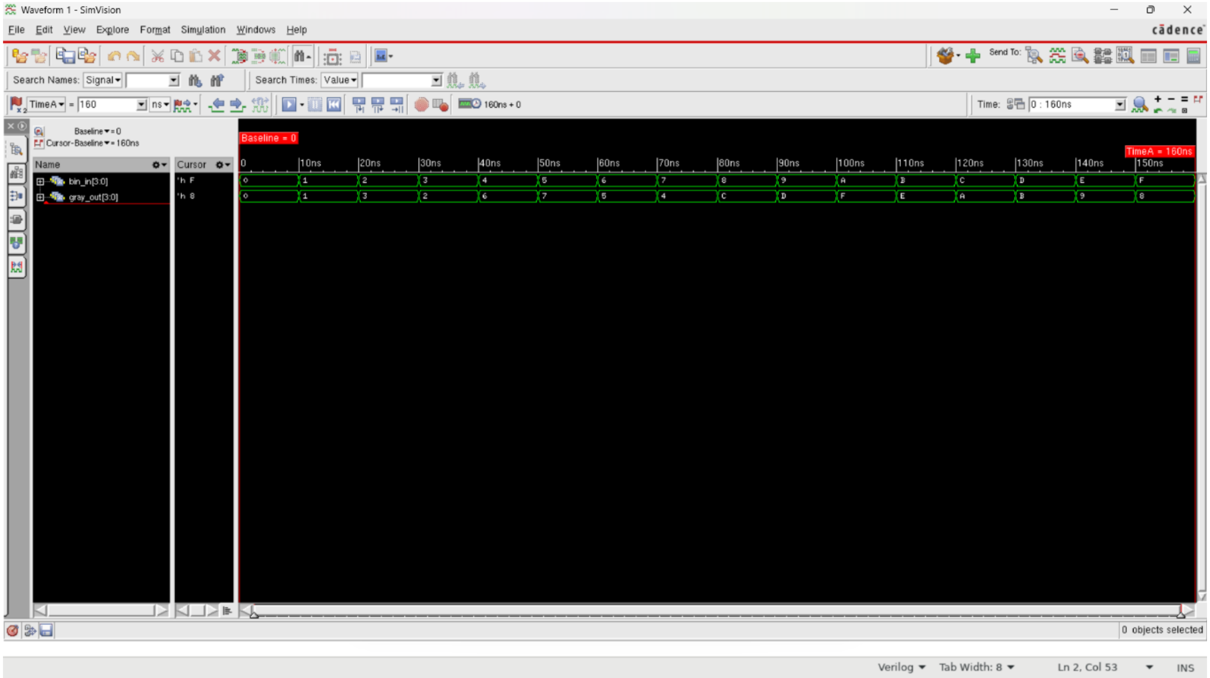
◆ **1.2) Test Bench Code**

```verilog
//TB CODE OF BINARY TO GRAY CONVERSION USING VERILOG
`timescale 1ns/1ps
module tb_bin_gray;
  reg [3:0]bin_in;
  wire [3:0]gray_out;

  bin_gray u1(.bin_in(bin_in),.gray_out(gray_out));
  initial begin
  $display("Time\tBinary\tGray");
  $monitor("%0dns\t%b\t%b", $time, bin_in, gray_out);


  bin_in = 4'b0000;#10;
  bin_in = 4'b0001;#10;
  bin_in = 4'b0010;#10;
  bin_in = 4'b0011;#10;
  bin_in = 4'b0100;#10;
  bin_in = 4'b0101;#10;
  bin_in = 4'b0110;#10;
  bin_in = 4'b0111;#10;
  bin_in = 4'b1000;#10;
  bin_in = 4'b1001;#10;
  bin_in = 4'b1010;#10;
  bin_in = 4'b1011;#10;
  bin_in = 4'b1100;#10;
  bin_in = 4'b1101;#10;
  bin_in = 4'b1110;#10;
  bin_in = 4'b1111;#10;

  $finish;
  end
endmodule
```

- **1.3) Schematic**



- **1.4) Wave Forms**



- Applications
  - Digital counters: Prevents glitches during state transitions.
  - Rotary encoders: Position sensors use Gray code to avoid ambiguity.
  - Communication systems: Prevents multiple-bit errors.
  - Error detection/correction: Useful in data transmission.