SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
College of Engineering and Technology
School of Electrical and Electronics
Department of Electronics and Communication Engineering

21ECC311L - VLSI Design Laboratory
V Semester, 2024-2025 (ODD Semester)

Title of Mini Project : GREYCODE COUNTER

Date of Submission :09.10.2024

Particulars	**Max. Marks**	**Marks Obtained**	
		Name:Ramprasaath R N	**Name:Pragadeshwaran G**
		Register No:RA2211004010417	**Register No:RA2211004010404**
Novelty	5		
Design Code	15		
Demo verification	10		
Viva	5		
Project Report	05		
Total	**40**		

REPORT VERIFICATION

Staff Name : Dr.K.Suganthi
Signature :

GREYCODE COUNTER

1. Objective

The general objective of a Gray code counter is to provide a counting mechanism that minimizes the number of bit changes between successive values, enhancing reliability and reducing the likelihood of errors in digital systems. By ensuring that only one bit changes at a time, Gray code counters are particularly effective in applications like rotary encoders, position tracking, and digital communication, where accurate state representation is critical.

2. Software Details

Computer with XILINIX ISE 7i.1 Software Specifications: HP Computer Ryzen5 Processor-2.8 GHz, 8GB RAM Softwares: XILINIX ISE 7i.1

3. Abstract /Introduction

This paper presents a Gray code counter designed to enhance reliability in digital systems by ensuring only a single bit changes between successive counts. This approach minimizes transition errors, making it ideal for applications such as rotary encoders and digital communication. The Gray code counter offers a robust solution with simplified hardware implementation and increased noise immunity, facilitating accurate state tracking in various electronic applications.

4. Code

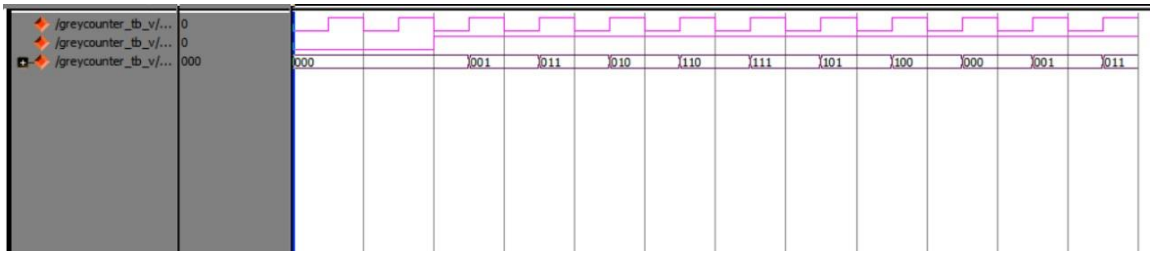
```
module gray_counter(
input clock,
input reset,
output reg [2:0] gray_out
);

reg [2:0] bin_counter;

always @(posedge clock or negedge reset) begin
if(~reset)
bin_counter <= 0;
else
bin_counter <= bin_counter + 1;
end

always @(bin_counter) begin
case(bin_counter)
3'b000 : gray_out = 3'b000;
3'b001 : gray_out = 3'b001;
3'b010 : gray_out = 3'b011;
3'b011 : gray_out = 3'b010;
3'b100 : gray_out = 3'b110;
3'b101 : gray_out = 3'b111;
3'b110 : gray_out = 3'b101;
3'b111 : gray_out = 3'b100;
endcase
end
```

```
endmodule
```

5.Test Bench:

```
module greycounter_tb_v;

        // Inputs
        reg clock;
        reg reset;

        // Outputs
        wire [2:0] gray_out;

        // Instantiate the Unit Under Test (UUT)
        greycounter uut (
                .clock(clock),
                .reset(reset),
                .gray_out(gray_out)
        );
         always #5 clock = ~clock;
        initial begin


                // Wait 100 ns for global reset to finish
         clock = 0;
  reset = 0;

  // Apply reset
  #10 reset = 0;  // Apply active low reset
  #10 reset = 1;  // De-assert reset

  // Monitor signals
  $monitor("Time=%d, clock=%b, reset=%b, gray_out=%b", $time, clock, reset, gray_out);

  // Run simulation for 100 time units
  #100;

  // Finish simulation
  $stop;
        end

endmodule
```

Output:



6.Result

A Gray code counter is a digital circuit that generates a sequence of Gray code, where consecutive numbers differ by only one bit. This makes it particularly useful in applications where minimizing errors is critical, such as in rotary encoders, analog-to-digital conversion, and error correction in communication systems. The counter works by converting binary values into Gray code, typically using an XOR logic operation, where each bit of the Gray code is generated by XOR-ing consecutive bits of the binary number.

7.Conclusion

In conclusion, the **Gray code counter** project demonstrates the effectiveness of Gray code in minimizing transition errors by ensuring that only one bit changes between consecutive states. This characteristic is particularly valuable in digital systems involving data conversion, rotary encoders, and noise-sensitive environments.