Digital Design Lab Report

Name	Dev Goel
Roll No.	B20CS090
Experiment Number	**II**

Objective

(i) Write a program to implement (with test benches).
  a.) Binary to Gray Code Converter.
  b.) Gray code to Binary code Converter.

(ii) Write a program to implement a BCD to 7-segment display decoder.

(iii) Implement a Buzzer system in Verilog. Write test bench and verify various cases.

Work 1a: Write a program to implement Binary to Gray Code converter. Write its test bench and simulate.

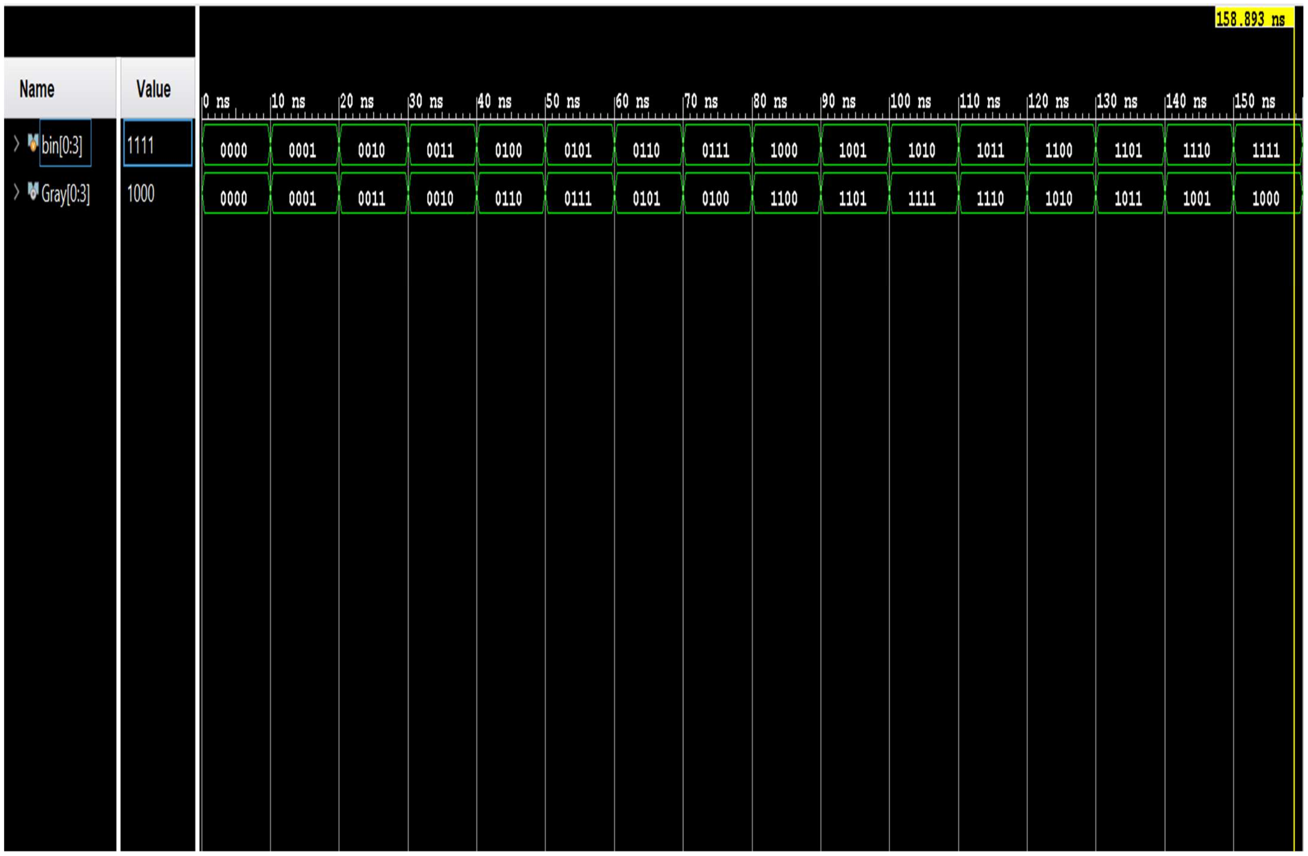
- **Binary to Gray Code**

```
`timescale 1ns / 1ps

module binary_to_gray(input [0:3] bin, output [0:3] Gray);

assign Gray[0] = bin[0];
assign Gray[1] = bin[0] ^ bin[1];
assign Gray[2] = bin[1] ^ bin[2];
assign Gray[3] = bin[2] ^ bin[3];

endmodule
```

- **Test Bench**

```
`timescale 1ns / 1ps

module test_binary_to_gray;

reg [0:3] bin;
wire [0:3] Gray;

binary_to_gray b2g(bin, Gray);

always
begin
    bin = 0;
    #10 bin = 1;
    #10 bin = 2;
    #10 bin = 3;
    #10 bin = 4;
    #10 bin = 5;
    #10 bin = 6;
    #10 bin = 7;
    #10 bin = 8;
    #10 bin = 9;
    #10 bin = 10;
    #10 bin = 11;
    #10 bin = 12;
    #10 bin = 13;
    #10 bin = 14;
    #10 bin = 15;
    #10;
   $stop;
  end
endmodule
```

- **Waveform**


158.893 ns
Name
Value
bin[0:3]
1111
Gray[0:3]
1000
0 ns
10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
70 ns
80 ns
90 ns
100 ns
110 ns
120 ns
130 ns
140 ns
150 ns
0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
0000 0001 0011 0010 0110 0111 0101 0100 1100 1101 1111 1110 1010 1011 1001 1000

<u>Work 1b:</u> Write a program to implement Gray Code to Binary converter. Write its test bench and simulate.

- **Gray Code to Binary**

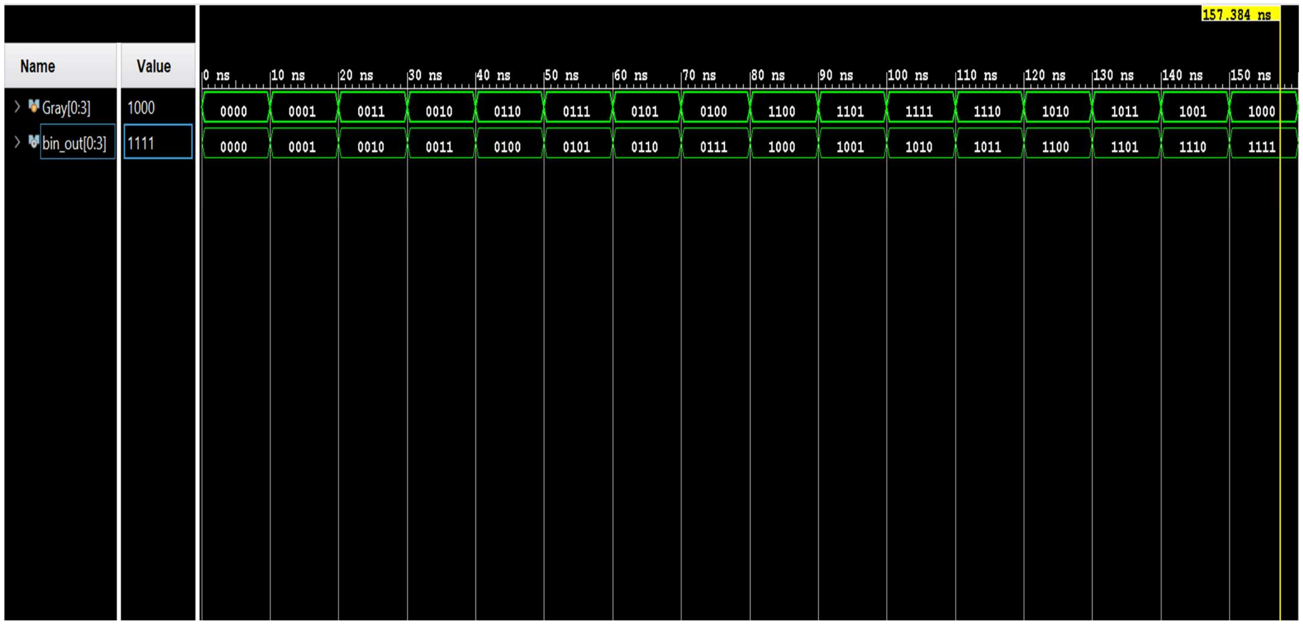
```
`timescale 1ns / 1ps

module gray_to_binary(input [0:3] G, output [0:3] bin);

assign bin[0] = G[0];
assign bin[1] = G[0] ^ G[1];
assign bin[2] = G[0] ^ G[1] ^ G[2];
assign bin[3] = G[0] ^ G[1] ^ G[2] ^ G[3];

endmodule
```

- **Test Bench**

```
`timescale 1ns / 1ps

module test_gray_to_binary;

reg [0:3] Gray;
wire [0:3] bin_out;

gray_to_binary g2b(Gray, bin_out);

always
begin
   Gray = 0;
   #10 Gray = 1;
```

```
    #10 Gray = 3;
    #10 Gray = 2;
    #10 Gray = 6;
    #10 Gray = 7;
    #10 Gray = 5;
    #10 Gray = 4;
    #10 Gray = 12;
    #10 Gray = 13;
    #10 Gray = 15;
    #10 Gray = 14;
    #10 Gray = 10;
    #10 Gray = 11;
    #10 Gray = 9;
    #10 Gray = 8;
    #10;
   $stop;
  end
endmodule
```

- **Waveform**



<u>Work 2:</u> Write a program to implement a BCD to 7-segment display decoder.

- **Four Bit Adder**

```
`timescale 1ns / 1ps
module bcd_to_seven(bcd, sevenSeg);

input [0:3] bcd;
output [0:6] sevenSeg;
reg [0:6] sevenSeg;

always @(bcd)
   begin
      case (bcd)
         0 : sevenSeg = 7'b1111110;
         1 : sevenSeg = 7'b0110000;
         2 : sevenSeg = 7'b1101101;
         3 : sevenSeg = 7'b1111001;
         4 : sevenSeg = 7'b0110011;
         5 : sevenSeg = 7'b1011011;
         6 : sevenSeg = 7'b1011111;
         7 : sevenSeg = 7'b1110000;
         8 : sevenSeg = 7'b1111111;
         9 : sevenSeg = 7'b1111011;
         default : sevenSeg = 7'b0000000;
      endcase
   end

endmodule
```

<u>Work 3:</u> Implement a Buzzer system in Verilog. Write test bench and verify various cases.

- **Buzzer**

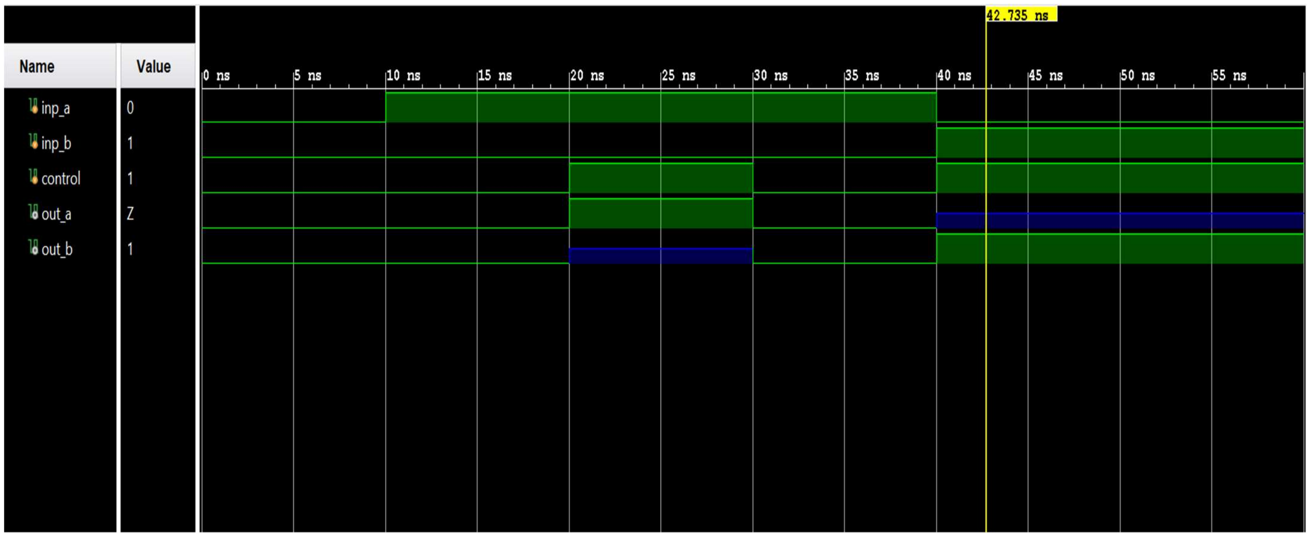
```
`timescale 1ns / 1ps
module buzzer(inp_a, inp_b, control, out_a, out_b);

input inp_a, inp_b, control;
output out_a, out_b;
wire p, q;

assign p = inp_a && control;
assign q = inp_b && control;

bufif0(out_a, p, q);
bufif0(out_b, q, p);

endmodule
```

- **Test Bench**

```
module test_buzzer();

reg inp_a, inp_b, control;
wire out_a, out_b;

buzzer bz(inp_a, inp_b, control, out_a, out_b);
initial
   begin
```

```
        inp_a = 0; inp_b = 0; control = 0;
        #10 inp_a = 1; inp_b = 0; control = 0;
        #10 inp_a = 1; inp_b = 0; control = 1;
        #10 inp_a = 1; inp_b = 0; control = 0;
        #10 inp_a = 0; inp_b = 1; control = 1;
    end
initial #60 $finish;
endmodule
```

- **Waveform**


42.735 ns
Name
Value
inp_a 0
inp_b 1
control 1
out_a Z
out_b 1
0 ns
5 ns
10 ns
15 ns
20 ns
25 ns
30 ns
35 ns
40 ns
45 ns
50 ns
55 ns