Verilog Code (Dice Game):-

```
`timescale 1ns / 1ps

module DiceGame (Rb, Reset, CLK, Sum, Roll, Win, Lose);

    input Rb, Reset, CLK;
    input [3:0] Sum;
    output reg Roll, Win, Lose;  // <-- FIXED: added reg

    reg [2:0] State, Nextstate;
    reg [3:0] Point;
    reg Sp;

    // State encoding for readability
    localparam IDLE  = 3'd0,
               ROLL1 = 3'd1,
               WIN   = 3'd2,
               LOSE  = 3'd3,
               WAIT  = 3'd4,
               ROLL2 = 3'd5;

    // Next State and Output Logic
    always @(*) begin
        // Default assignments
        Nextstate = State;
        Roll = 1'b0;
        Win = 1'b0;
```

```
        Lose = 1'b0;
        Sp = 1'b0;

        case (State)
            IDLE: begin
                if (Rb)
                    Nextstate = ROLL1;
            end

            ROLL1: begin
                Roll = 1'b1;
                if (Sum == 7 || Sum == 11)
                    Nextstate = WIN;
                else if (Sum == 2 || Sum == 3 || Sum == 12)
                    Nextstate = LOSE;
                else begin
                    Sp = 1'b1;
                    Nextstate = WAIT;
                end
            end

            WIN: begin
                Win = 1'b1;
                if (Reset)
```

```
                    Nextstate = IDLE;
            end

            LOSE: begin
                Lose = 1'b1;
                if (Reset)
                    Nextstate = IDLE;
            end

            WAIT: begin
                if (Rb)
                    Nextstate = ROLL2;
            end

            ROLL2: begin
                Roll = 1'b1;
                if (Sum == Point)
                    Nextstate = WIN;
                else if (Sum == 7)
                    Nextstate = LOSE;
                else
                    Nextstate = WAIT;
            end

```

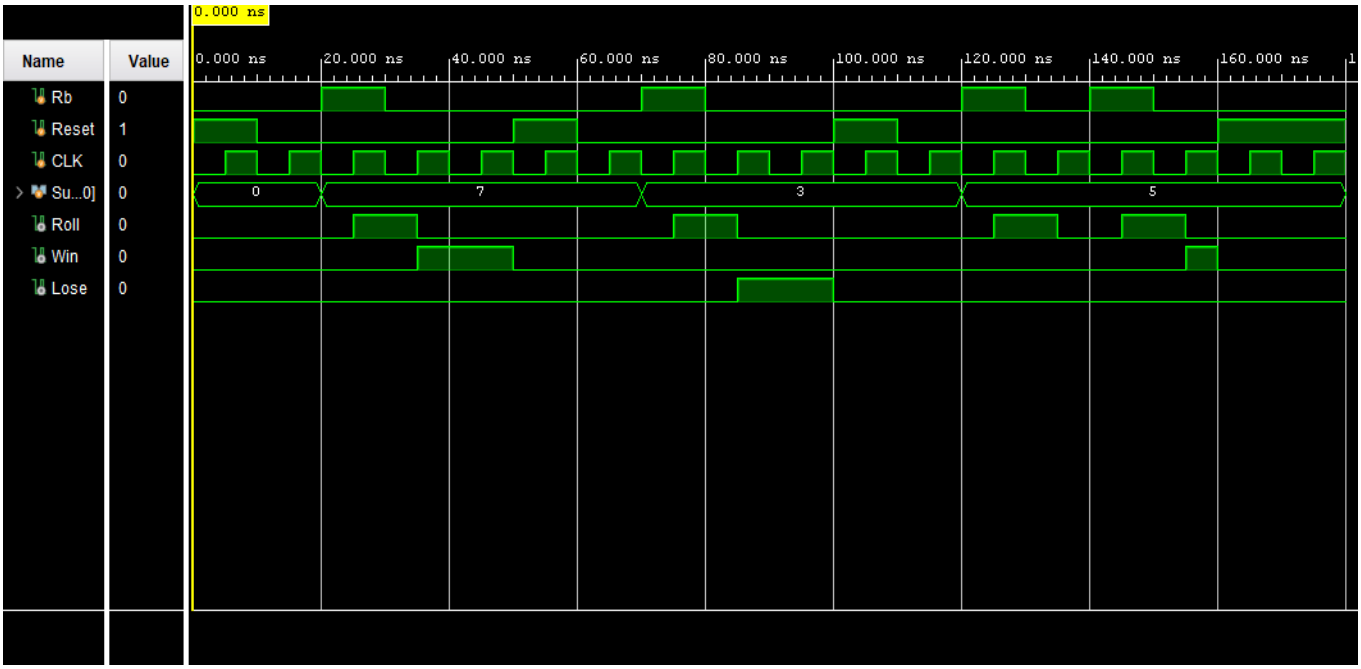
```
            default: Nextstate = IDLE;
        endcase
    end

    // State and Point Update
    always @(posedge CLK or posedge Reset) begin
        if (Reset) begin
            State <= IDLE;
            Point <= 4'd0;
        end else begin
            State <= Nextstate;
            if (Sp)
                Point <= Sum;
        end
    end

endmodule
```

Test Bench (Dice Game):-

```
module game_sim;

    reg Rb, Reset, CLK;
    reg [3:0] Sum;
    wire Roll, Win, Lose;

    // Instantiate the DiceGame module
    DiceGame uut (
        .Rb(Rb),
        .Reset(Reset),
        .CLK(CLK),
        .Sum(Sum),
        .Roll(Roll),
        .Win(Win),
        .Lose(Lose)
    );

    // Clock generation
    always #5 CLK = ~CLK;

    initial begin
        // Initialize inputs
        CLK = 0;
        Reset = 1;
        Rb = 0;
        Sum = 0;
```

```

        #10 Reset = 0;
        #10 Rb = 1; Sum = 7; // First roll - win
        #10 Rb = 0;
        #20 Reset = 1;       // Reset
        #10 Reset = 0;

        #10 Rb = 1; Sum = 3; // First roll - lose
        #10 Rb = 0;
        #20 Reset = 1;       // Reset
        #10 Reset = 0;

        #10 Rb = 1; Sum = 5; // First roll - go to WAIT
        #10 Rb = 0;
        #10 Rb = 1; Sum = 5; // Matching point - win
        #10 Rb = 0;
        #10 Reset = 1;

        #20 $finish;
    end

endmodule
```

Simulation Result:-



Project Summary:-

Settings Edit

Project name:	dice_game
Project location:	C:/Users/www.com/dice_game
Product family:	Zynq UltraScale+ RFSoCs
Project part:	Xilinx Zynq UltraScale+ RFSoC ZCU1275 Characterization Kit (xczu29dr-ffvf1760-2-e)
Top module name:	DiceGame
Target language:	Verilog
Simulator language:	Mixed
Target Simulator:	Vivado Simulator

Board Part

Display name:	Xilinx Zynq UltraScale+ RFSoC ZCU1275 Characterization Kit
Board part name:	xilinx.com:zcu1275:part0:1.0
Board revision:	2.0
Connectors:	No connections
Repository path:	E:/Xilinx/Vivado/2024.2/data/xhub/boards
URL:	https://www.xilinx.com/products/boards-and-kits/zcu1275.html
Board overview:	Xilinx Zynq UltraScale+ RFSoC ZCU1275 Characterization Kit