```
`timescale 1ns / 1ps


module GameState(
    input clk, btnU, TimeUp, //(8 Seconds)
        GameCountdown,
        Win,

    output Chill, Go, Play,
        ShowMem,
        ShowCountdown,
        CountingDown,
        LDTime,
        FlashTime,
        FlashBorder,
    // output [4:0] NS,
    output [4:0] PS
    );
    wire [4:0] NS;
    // wire [4:0] PS;

        // Chill
    assign NS[0] = (PS[0] & ~btnU) | (PS[3] & btnU) | (PS[4] & btnU);
        // Go (lasts 8 seconds long) -  Flash Borders, Load/show timer, Move balls
    assign NS[1] = (PS[0] & btnU) | ( PS[1] & ~TimeUp);
        // Play (after 8 seconds) -
    assign NS[2] = (PS[1] & TimeUp) | (PS[2] & ~GameCountdown & ~Win);
        // Win when blue and red balls are sepparated
    assign NS[3] = (PS[2] & Win) | (PS[3] & ~btnU);
        // Lose
    assign NS[4] = (PS[2] & GameCountdown & ~Win) | (PS[4] & ~btnU);

        assign Go = (PS[0] & btnU);
        assign ShowMem = (~PS[0] & ~PS[1]);
        assign CountingDown = (PS[0] & btnU);
        assign ShowCountdown = PS[2];
        assign Play = PS[2];
        assign FlashTime = PS[1] | PS[4];
        assign FlashBorder = PS[1] | (Win & ~btnU);

//  FlipFlops to store ball position
    FDRE #(.INIT(1'b1)) BallStorage_0 (.C(clk), .CE(1'b1), .R(1'b0), .D(NS[0]),
.Q(PS[0]));        //Start State
    FDRE #(.INIT(1'b0)) BallStorage_1to4 [4:1] (.C({4{clk}}), .R({4{1'b0}}),
.CE({4{1'b1}}), .D(NS[4:1]), .Q(PS[4:1]));
```

```
endmodule
```