

题目：电子骰子

在SystemVerilog中，你可以使用线性反馈移位寄存器（LFSR）来实现一个随机数生成器。以下是一个简单的LFSR实现，它使用了always块来在时钟边沿更新寄存器的值，并使用了异或运算来生成新的位。

```
module LFSR_random_generator(
    input wire clk,
    input wire reset,
    output wire [7:0] random_out
);

// LFSR的反馈多项式为  $x^8 + x^6 + x^5 + x^4 + 1$ 
reg [7:0] lfsr;
always_ff @(posedge clk or posedge reset) begin
    if (reset) begin
        // 当复位信号为高时，将LFSR重置
        lfsr <= 8'b1;
    end else begin
        // 计算新的LFSR值
        lfsr <= {lfsr[6:0], lfsr[7] ^ lfsr[5] ^ lfsr[4] ^ lfsr[3]};
    end
end

// 将LFSR的值输出作为随机数
assign random_out = lfsr;

endmodule
```

要求1：三组随机骰子动态显示

将8位信号 random_out 按以下规则划分为三组：

- 骰子1：高位段 [7:6]（取值范围0~3），分配至第一个数码管，译码后显示对应点数，未使用的数码管高位输入端口固定接地（逻辑低电平）。
 - 骰子2：中位段 [5:3]（取值范围0~7），分配至第二个数码管，未使用的最高位端口接地。
 - 骰子3：低位段 [2:0]（取值范围0~7），分配至第三个数码管，未使用的最高位端口接地。
- 三组骰子数值每1秒刷新一次，实现持续随机显示效果。
 - 复位按下时数码管显示0，抬起后恢复到随机显示状态。
 - 复位信号有效时，所有数码管强制显示0点；复位释放后，立即恢复随机刷新。

要求2：单次按键锁定与结果判定

- 新增一个独立按键，当按键按下并释放后（上升沿触发）：立即停止所有数码管的刷新，锁定当前显示值；

- 计算三组骰子点数的总和，若总和 ≥ 10 ，点亮LED；否则保持LED熄灭。
- 复位信号有效时，除保持要求1中的显示0点，还需强制熄灭LED。

要求3：分步按键

每次按下并释放按键（上升沿触发）时，按顺序执行以下操作：

- 第1次按键：锁定骰子1的当前值，骰子2、3继续随机刷新；
- 第2次按键：锁定骰子2的当前值，仅骰子3继续刷新；
- 第3次按键：锁定骰子3的当前值，骰子3停止刷新，立即计算三组值总和：若总和 ≥ 10 ，点亮LED；否则熄灭LED。
- 复位信号有效时：清除所有锁定状态，恢复三组骰子的随机刷新；强制熄灭LED，等待下一次按键触发。