

# 计算机组成原理 · 实验2报告

计01 容逸朗 2020010869

## 实验过程

首先在 `lab2_top.sv` 加入如下代码，这段代码由上以下分别绑定了计数器、按键检测和译码器模块的接口。

```
1 // 内部信号声明
2 logic trigger;
3 logic [3:0] count;
4
5 // 计数器模块
6 counter u_counter (
7     .clk      (clk_10M),
8     .reset    (reset_of_clk10M),
9     .trigger  (trigger),
10    .count    (count)
11 );
12
13 // 按键检测模块，在按键上升沿（按下）后输出高电平脉冲
14 button_in u_button_in (
15     .clk      (clk_10M),
16     .reset    (reset_of_clk10M),
17     .push_btn (push_btn),
18     .trigger  (trigger)
19 );
20
21 // 低位数码管译码器
22 SEG7_LUT u_seg (
23     .oSEG1(dpy0),
24     .iDIG (count)
25 ); // dpy0 是低位数码管
```

计数器模块当 `reset` 为 1 时输出置零，否则按照约定把输出值加 1，直至计数到达 0xF 为止。

```
1 reg [3:0] count_reg;
2 always_ff @ (posedge clk or posedge reset) begin
3     if (reset) begin
4         count_reg <= 4'd0;
5     end else begin
6         if (trigger && count_reg ≠ 4'hf) // 增加此处
7             count_reg <= count_reg + 4'd1; // 暂时忽略计数溢出
8     end
9 end
10 assign count = count_reg;
```

按键检测模块需要记忆按键在上一时刻的值，这样可以保证每次按键时数值只会增加 1。

```
1 reg last_stat_reg;
2 reg trigger_reg;
3
4 always_ff @ (posedge clk or posedge reset) begin
5     if (reset) begin
6         trigger_reg <= 1'b0;
7         last_stat_reg <= 1'b0;
8     end else begin
9         if (push_btn) begin
10            if (!last_stat_reg) begin
11                trigger_reg <= 1'b1;
12                last_stat_reg <= 1'b1;
13            end else
14                trigger_reg <= 1'b0;
15        end else begin
16            last_stat_reg <= 1'b0;
17            trigger_reg <= 1'b0;
18        end
19    end
20 end
21
22 assign trigger = trigger_reg;
```

## 思考题

1. 计数器模块中提到的异步逻辑与同步逻辑有何不同？可以通过观察 Vivado 综合后的电路原理图，并且查阅相关资料回答本题。

同步逻辑的时钟之间有固定的因果关系，这是由于同步逻辑电路利用一个全局时钟脉冲信号，当此信号同时到达元件时才会引起状态改变。而异步逻辑没有全局统一的时钟驱动（即时钟之间无固定的因果关系），任意输入信号都可以引起电路状态改变，因此电路中状态变化的时刻是不稳定的。