

流水灯

代码

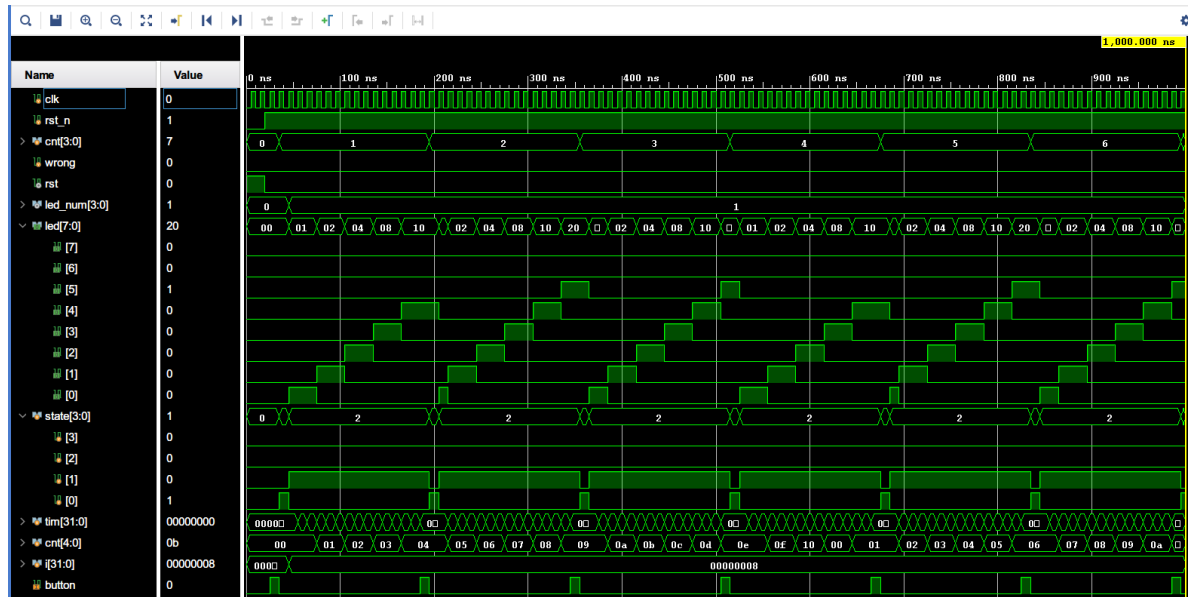
```
1 // Author: Chiro, Date: 2021/11/10
2 module flowing_water_lights (input wire clk,
3                               input wire rst,
4                               input wire button,
5                               output reg [7:0] led);
6     // parameter delay      = 32'd2;
7     // 分频数量
8     parameter delay      = 32'd100000000;
9     parameter STATE_PRE  = 0;
10    parameter STATE_INIT = 1;
11    parameter STATE_RUN  = 2;
12    reg [3:0] state;
13    reg [31:0] tim;
14    reg [4:0] cnt;
15    integer i;
16
17    always @ (posedge clk or posedge rst) begin
18        if (rst) begin
19            led    <= 8'b0;
20            cnt    <= 5'b0;
21            tim    <= 32'b0;
22            i      <= 0;
23            // 初始化当前状态机为开始之前
24            state <= STATE_PRE;
25        end
26        else begin
27            if (state == STATE_PRE) begin
28                // 开始之前状态，等待按键开始，就进入初始化状态
29                if (button)
30                    state <= STATE_INIT;
31            end
32            else if (state == STATE_INIT) begin
33                // 初始化状态，设置相关信息之后进入运行状态
34                for (i = 0; i < 8; i = i + 1)
35                    led[i] = (i == 0) ? 1'b1 : 1'b0;
36                state <= STATE_RUN;
37            end
38            else if (state == STATE_RUN) begin
39                // 运行状态，如果有按键就重新初始化
40                if (button)
41                    state <= STATE_INIT;
42                else begin
43                    // 计数器溢出，更新cnt
44                    if (tim == delay) begin
45                        tim <= 32'b0;
46                        // 进行一个位的移
47                        led <= {led[6 -: 7], led[7]};
48                        cnt <= cnt + 5'b1;
49                        // cnt 溢出，回到初始状态
50                        if (cnt == 5'd16)
```

```

51         cnt <= 5'b0;
52     end else
53         // 计数
54         tim <= tim + 32'b1;
55     end
56 end
57 end
58 end
59 endmodule
60

```

波形分析



`flow_water_lights` 模块完成的是流水灯的功能，在根据时间进行灯的流水变化的同时还要考虑按键复位的功能。

1. 当 `rst` 拉高，系统复位
2. 当 `rst` 拉低，系统开始运行
3. 每隔2个时钟周期，`tim` 自增1；当 `tim` 达到 `delay` 表示 `tim` 计时完成，需要更新 `cnt` 再复位 `tim`
4. `cnt` 更新同时，`led` 位移一位
5. 同时，当 `button` 按下，位移复位，重新从第一位开始流水，`state` 变化为： `STATE_RUN -> STATE_INIT -> STATE_RUN`

节日彩灯

代码

```

1 // Author: Chiro, Date: 2021/11/10
2 module holiday_lights (input wire clk,
3                       input wire rst,
4                       input wire button,
5                       input wire [2:0] switch,
6                       output reg [15:0] led);
7
8     // 分频数量
9     // parameter delay      = 32'd2;
10    parameter delay      = 32'd100000000;
11    parameter STATE_PRE  = 0;
12    parameter STATE_INIT = 1;

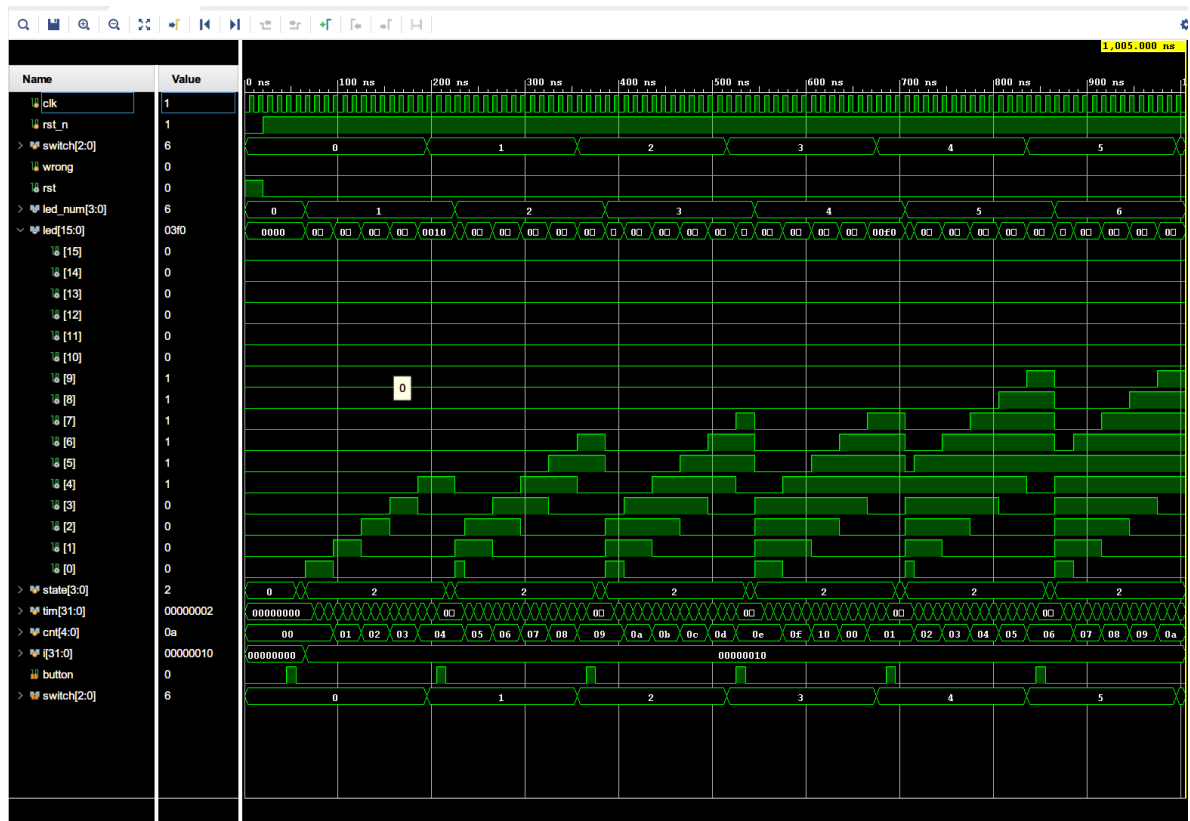
```

```

12     parameter STATE_RUN   = 2;
13     reg [3:0] state;
14     reg [31:0] tim;
15     reg [4:0] cnt;
16     integer i;
17
18     always @ (posedge clk or posedge rst) begin
19         if (rst) begin
20             led    <= 16'b0;
21             cnt    <= 5'b0;
22             tim    <= 32'b0;
23             i      <= 0;
24             // 初始化当前状态机为开始之前
25             state <= STATE_PRE;
26         end
27         else begin
28             if (state == STATE_PRE) begin
29                 // 开始之前状态，等待按键开始，就进入初始化状态
30                 if (button)
31                     state <= STATE_INIT;
32             end
33             else if (state == STATE_INIT) begin
34                 // 初始化状态，设置相关信息之后进入运行状态
35                 for (i = 0; i < 16; i = i + 1)
36                     led[i] = (i < switch + 1) ? 1'b1 : 1'b0;
37                 state <= STATE_RUN;
38             end
39             else if (state == STATE_RUN) begin
40                 // 运行状态，如果有按键就重新初始化
41                 if (button)
42                     state <= STATE_INIT;
43                 else begin
44                     // 计数器溢出，更新cnt
45                     if (tim == delay) begin
46                         tim <= 32'b0;
47                         // 进行一个位的移
48                         led <= {led[14 -: 15], led[15]};
49                         cnt <= cnt + 5'b1;
50                         // cnt 溢出，回到初始状态
51                         if (cnt == 5'd16)
52                             cnt <= 5'b0;
53                     end else
54                         // 计数
55                         tim <= tim + 32'b1;
56                 end
57             end
58         end
59     end
60 endmodule
61

```

波形分析



`holiday_lights` 模块完成的是流水灯的功能，在根据时间进行灯的流水变化的同时还要考虑按键复位的功能。

1. 当 `rst` 拉高，系统复位
2. 当 `rst` 拉低，系统开始运行，`state = STATE_PRE`，等待 `button` 按下
3. 每隔2个时钟周期，`tim` 自增1；当 `tim` 达到 `delay` 表示 `tim` 计时完成，需要更新 `cnt` 再复位 `tim`
4. `cnt` 更新同时，`led` 位移一位
5. 同时，当 `button` 按下，位移复位，重新从第一位开始流水，`state` 变化为：`STATE_RUN -> STATE_INIT -> STATE_RUN`