

目录

1	实验原理	2
2	实验内容	2
2.1	实验步骤	2
2.2	实验结果	3
2.3	拓展实验	3

集成电路实验一

数字系统综合实验-Quartus II、Modelsim、FPGA

操作入门

第一组物理 2001 孙陶庵 202011010101 2023/12/05 物电院 A116 实验

2023 年 12 月 7 日

1 实验原理

2 实验内容

2.1 实验步骤

使用 Quartus II 编程，写出一份可以操控电路板上 led 灯的程序，并调控参数观察现象。

实验步骤：

1. 电脑创建程序写出程序
2. 编译，debug
3. 使用 Modelsim 仿真并观察波形，确认无误后将电路板接进电脑
4. 使用 Quartus II 将程序导入电路板

2.2 实验结果

2.3 拓展实验

读懂代码，试修改代码，完成以下实验步骤。

- a. 如何改变流水灯的移动速度？
- b. 如何改变流水灯的移动方向？
- c. 如何移动两个 LED 灯？
- d. 分别用 *Key_1*、*Key_2*、*Key_3* 三个按键来控制移动频率 1Hz、2Hz 和移动方向
- e. 查看资源占有率及 RTL 视图

拓展实验结果：

- a.LED 的移动速度是通过计时器 *counter* 和计时结束标识 *end_cnt* 来控制的。所以可以透过修改这一部分来改变移动速度

```
1      always @(posedge clk or negedge rst_n or negedge key_1 or negedge key_2
2      )
3
4      if (!rst_n)
5          counter <= 0;
6          end_cnt <= END_CNT_1S; // 初始设置为1秒的计时结束标识
7
8      else if (!key_1) // 按下后周期变为1S
9          end_cnt <= END_CNT_1S;
10
11     else if (!key_2) // 按下后周期变为2S
12         end_cnt <= END_CNT_2S;
13
14     else if (counter == end_cnt)
15         counter <= 0;
16
17     else
18         counter <= counter + 1'b1;
```

通过按下 *Key_1*，LED 的速度将被设置为 1 秒一次，而按下 *Key_2*，LED 的速度将被设置为 2 秒一次。这是因为 *end_cnt* 在这里用于确定计时器何时达到结束状态，

进而改变 LED 的状态。

因此，如果希望改变 LED 的移动速度，可以通过按下不同的按键来选择不同的计时结束标识。

b.LED 的移动方向是通过按键输入 *key_3* 控制的。按下 *key_3* 时，*direction* 寄存器会取反，从而改变 LED 的移动方向。

```
1
2 always @(posedge clk or negedge rst_n or negedge key_1 or negedge key_2 or
   negedge key_3)
3     if (!rst_n)
4     begin
5         counter <= 0;
6         end_cnt <= END_CNT_1S; // 初始设置为1秒的计时结束标识
7         direction <= 1'b0; // 初始方向为0
8     end
9     else if (!key_1) // 按下后周期变为1S
10        end_cnt <= END_CNT_1S;
11    else if (!key_2) // 按下后周期变为2S
12        end_cnt <= END_CNT_2S;
13    else if (!key_3) // 按下后改变方向
14        direction <= ~direction; // 取反方向
15    else if (counter == end_cnt)
16        counter <= 0;
17    else
18        counter <= counter + 1'b1;
```

当按下 *key_3* 时，*direction* 的值将翻转，从而改变 LED 的移动方向。这是通过 *direction* <= *direction*; 这一语句实现的。

因此，如果希望改变 LED 的移动方向，只需按下 *key_3* 即可。

c. 可以尝试使用两个计时器和两个计时结束标识来分别控制两个 LED 的移动。

```
1
2 module dual_led (
3     input wire clk,
4     input wire rst_n,
5     input wire key_1,
6     input wire key_2,
7     input wire key_3,
8     output reg[3:0] led1,
9     output reg[3:0] led2
10 );
11
12 reg [25:0] counter1;    // 1秒计数器1
13 reg [25:0] counter2;    // 1秒计数器2
14 reg [25:0] end_cnt1;    // 结束时间周期1
15 reg [25:0] end_cnt2;    // 结束时间周期2
16 reg direction1;        // 方向1, 0左1右
17 reg direction2;        // 方向2, 0左1右
18
19 parameter END_CNT_1S = 4999999; // 1秒计数结束标识
20 parameter END_CNT_2S = 9999999; // 2秒计数结束标识
21
22 always @(posedge clk or negedge rst_n or negedge key_1 or negedge key_2 or
23     negedge key_3)
24     begin
25         counter1 <= 0;
26         end_cnt1 <= END_CNT_1S;
```

```

27     direction1 <= 1'b0;  // 初始方向为0
28
29     end
30
31     else if (!key_1)
32         end_cnt1 <= END_CNT_1S;
33
34     else if (!key_2)
35         end_cnt1 <= END_CNT_2S;
36
37     else if (!key_3)
38         direction1 <= ~direction1;
39
40
41
42 // Counter1 logic
43
44 if (counter1 == end_cnt1)
45     counter1 <= 0;
46
47 else
48     counter1 <= counter1 + 1'b1;
49
50
51
52 always @(posedge clk or negedge rst_n or negedge key_1 or negedge key_2 or
53 negedge key_3)
54
55     if (!rst_n)
56     begin
57         counter2 <= 0;
58
59         end_cnt2 <= END_CNT_1S;
60
61         direction2 <= 1'b0;  // 初始方向为0
62
63     end
64
65     else if (!key_1)
66         end_cnt2 <= END_CNT_1S;
67
68     else if (!key_2)
69         end_cnt2 <= END_CNT_2S;
70
71     else if (!key_3)
72         direction2 <= ~direction2;

```

```

55
56 // Counter2 logic
57 if (counter2 == end_cnt2)
58     counter2 <= 0;
59 else
60     counter2 <= counter2 + 1'b1;
61
62 always @(posedge clk or negedge rst_n)
63 begin
64     if (!rst_n)
65     begin
66         led1 <= 4'b0001;
67         led2 <= 4'b0001;
68     end
69     else
70     begin
71         // LED1 logic
72         case({counter1 == end_cnt1, direction1})
73             2'b10: led1 <= led1 << 1;
74             2'b01: led1 <= led1 >> 1;
75             default: led1 <= led1;
76         endcase
77
78         // LED2 logic
79         case({counter2 == end_cnt2, direction2})
80             2'b10: led2 <= led2 << 1;
81             2'b01: led2 <= led2 >> 1;
82             default: led2 <= led2;
83         endcase

```

```
84     end
85 end
86
87 endmodule
```

分别由两个计时器 counter1 和 counter2 控制。按键 *key_3* 仍然用于改变方向，而按键 *key_1* 和 *key_2* 分别用于选择 1 秒和 2 秒的计时结束标识。LED 的移动逻辑分别由两个独立的 case 语句控制。

d. *Key_1* 选择 1Hz 的计时结束标识和向左移动。*Key_2* 选择 2Hz 的计时结束标识和向左移动。*Key_3* 按下则切换移动方向。