代码附录

```
//1: 顶层模块================================================
 module top_module(
  input clk,
  input start,
  input reset,
  input key0,key1,key2,key3,
  output row,
  output [3:0]led,
  output [5:0]dig,
  output [7:0]seg,
  output beep,
  output [4:0]led_b);
    assign row=1'b0;
    wire clk_1k;
    wire clk_1s;
    wire clk_20ms;

  fenpin u1(.clk(clk),
         .clk_1k(clk_1k),
         .clk1sout(clk_1s),
         .clk_20ms(clk_20ms));

         wire k0;
         wire k1;
         wire k2;
         wire k3;
  key_debounce u2(
         .clk(clk_20ms),
         .key_in(key0),
         .key_out(k0));
  key_debounce u3(
         .clk(clk_20ms),
         .key_in(key1),
         .key_out(k1));
  key_debounce u4(
```

```
36          .clk(clk_20ms),
37          .key_in(key2),
38          .key_out(k2));
39    key_debounce u5(
40          .clk(clk_20ms),
41          .key_in(key3),
42          .key_out(k3));
43    wire [1:0]state;
44    wire [1:0]floor;
45    operation u6(
46          .clk(clk),
47          .set(start),
48          .reset(reset),
49          .key0(k0),
50          .key1(k1),
51          .key2(k2),
52          .key3(k3),
53          .led(led),
54          .state(state),
55          .floor(floor));
56    display u7(
57          .floor(floor),
58          .state(state),
59          .clk(clk_1k),
60          .seg(seg),
61          .clk_1s(clk_1s),
62          .led_b(led_b),
63          .dig(dig));
64    beep u8(
65          .clk(clk),
66          .state(state),
67          .floor(floor),
68          .beep(beep));
69  endmodule
```

```
70
71 //2.分频器================================================
72 module fenpin(
73     input clk,clr,
74     output reg clk_1k=0,  //1kHz信号
75     output reg clk1sout=0,//1Hz信号
76     output reg clk_20ms=0,//50Hz信号
77     integer    clk1s_cnt=0
78     );
79  reg[24:0] cnt=0;
80  always @ (posedge clk)//分频得到1kHz信号
81     begin
82       if (cnt==25000)
83         begin
84            clk_1k=~clk_1k; cnt=0;
85          end
86       else  cnt=cnt+1;
87     end
88
89 reg [31:0] temp = 0;
90 always@(posedge clk)//分频得到50Hz信号
91     begin
92       if(temp == 999999)
93       begin
94             temp <= 0;clk_20ms <= ~clk_20ms;
95       end
96       else  temp <= temp+1;
97       end
98
99  always@(posedge clk)//分频得到1Hz信号
100  begin
101     if(clr)
102     begin
103            clk1s_cnt<=0; clk1sout<=0;
```

```
104     end
105     else  if(clk1s_cnt==24999999)
106         begin
107             clk1s_cnt<=0;clk1sout<=~clk1sout;
108         end
109     else    clk1s_cnt<=clk1s_cnt+1;
110  end
111 endmodule
112 
113 //3.按键消抖模块=====================================
114 module key_debounce(
115     input key_in,
116     input clk,
117     output key_out);
118     reg  btn0=0;
119     reg  btn1=0;
120     reg  btn2=0;
121 
122 always@(posedge clk)
123   begin
124     btn0<=key_in;
125     btn1<=btn0;
126     btn2<=btn1;
127   end
128 
129 assign key_out=((btn0&btn1)&(~btn2) | (btn0&btn1&btn2) | ((~btn0)
    &btn1&btn2));
130 endmodule
131 
132 //4.状态控制模块=====================================
133 module operation(
134     input clk,
135     input set,
136     input reset,
```

```
    input key0,
    input key1,
    input key2,
    input key3,
    output reg [3:0]led=0,//指示灯
    output reg [1:0]state=0,//0待机状态; 1上行; 2下行
    output reg [1:0]floor=1  //1一楼; 2二楼
        );
    reg [27:0]cnt=0;
    reg [27:0]clk_count=0;
    reg k0=0;
    reg k1=0;
    reg k2=0;
    reg k3=0;
    reg [1:0] first=0;
always@(posedge clk)
  begin
    if(!reset)//复位开关被拨回
       begin
           if(floor==2)//如果电梯在2楼, 则运行五秒后回到一楼
             if(clk_count==249999999) //5s计时
             begin
                clk_count=0;
                floor=1;    //1楼
                state=0;    //待机状态
                led=4'b0000;//指示灯不亮
              end
             else
             begin
                clk_count=clk_count+1;
                state=2;    //下行状态
                led=4'b0000;//指示灯不亮
             end
       end
```

```
171     else if(set)//复位开关未被拨下，且启动start有效
172     begin
173       if(led)//当led非0，即电梯处于运行状态时
174         begin
175           if(clk_count==249999999)//5s计时
176            begin
177              clk_count=0;
178              if(floor==1)
179                  begin floor=floor+1;
180                  end//计时五秒后从一楼切换到二楼
181              else
182                  begin floor=floor-1;
183                  end//计时五秒后从二楼切换到一楼
184
185              //key3，key2接连被按下，到二楼后保持led2亮起，切换到
                    下行状态
186              if((k3&k2)&&first==1)
187                  begin state=2;k3=0;k2=0;led[3]=0;first=0;
188                  end
189
190              //key3，key0接连被按下，到二楼后保持led0亮起，切换到
                    下行状态
191              else if(k3&k0)
192                  begin state=2;k3=0;k0=0;led[3]=0;
193                  end
194
195              //key2，key3接连被按下，到一楼后保持led3亮起，切换到
                    上行状态
196              else if((k2&k3)&&first==2)
197                  begin state=1;k2=0;k3=0;led[2]=0;first=0;
198                  end
199
200              //key2，key1接连被按下，到一楼后保持led1亮起，切换到
                    上行状态
```

```
201             else if(k2&k1)
202                 begin state=1;k2=0;k1=0;led[2]=0;
203                 end
204
205             //回到待机状态
206             else
207                 begin state=0;led=4'b0000;k0=0;k1=0;k2=0;k3=0;
208                 end
209         end
210
211         else  clk_count=clk_count+1;
212     //在运行过程中保持指示灯正常亮起，并且在按键在电梯运行过程中被按下时进行记录
213      if(key2|k2)
214         begin k2=1;k0=0;led[2]=1;
215         end
216      if(key0|k0)
217         begin k0=1;k2=0;led[0]=1;
218         end
219      if(key3|k3)
220         begin k3=1;k1=0;led[3]=1;
221         end
222      if(key1|k1)
223         begin k1=1;k3=0;led[1]=1;
224         end
225     end
226
227     else if((key0)&&floor==2)  //当前在二楼，一楼按key0
228             begin
229                 led[0]=1;
230                 state=2; //下行
231             end
232
233     else if((key1)&&floor==1)  //当前在一楼，二楼按key1
```

```
234             begin
235                 led[1]=1;
236                 state=1;//上行
237             end
238 
239     else if((key2)&&floor==2)  //当前在二楼 电梯内按下
240             begin
241                 led[2]=1;
242                 state=2; //下行
243                 k2=1;
244                 first=2;//用于表明key2先于key3被按下
245             end
246 
247     else if((key3)&&floor==1)  //当前在一楼 电梯内按上
248             begin
249                 led[3]=1;
250                 state=1; //上行
251                 k3=1;
252                 first=1;//用于表明key3先于key2被按下
253             end
254         end
255     end
256 endmodule
257 
258 //5.译码显示及流水指示灯模块============================
259 
260 module display(
261     input [1:0]floor,//楼层
262     input [1:0]state,//状态
263     input clk,
264     input clk_1s,
265     output reg [4:0] led_b=0,
266     output reg [7:0] seg,
267     output reg [5:0] dig
```

```
);
reg num=0;
        always@(posedge clk)
        begin
                if (num==1) num=0;
                else num=num+1;
        end
   //位选
        always@(num)
        begin
                case(num)
                0:dig=6'b111110;
                1:dig=6'b111101;
                default: dig=0;
                endcase
        end

always@(posedge clk_1s or posedge state or negedge state)
begin
        if(state==0)
            led_b[4:0]=5'b00000;
    else if (state ==1)//电梯处于上行状态
        begin
            if (led_b[4:0]==5'b00000)
                led_b[4:0]=5'b10000;
        else
                led_b[4:0]=led_b[4:0]>>1;
    //向右移一位。10000-01000-00100-00010-00001-00000
        end
    else if (state ==2)//电梯处于下行状态
        begin
            if (led_b[4:0]==5'b00000)
                led_b[4:0]=5'b00001;//到了00000，下一个是00001
            else
```

```
                led_b[4:0]=led_b[4:0]<<1;
    //向左移一位。00001-00010-00100-01000-10000-00000
        end
    else led_b[4:0]=5'b00000;
        end

//选择器，确定显示数据
reg [3:0] disp_data;
always@(num)
begin
        case(num)
        0:disp_data=floor;
        1:disp_data=state+3'b011;//3待机 4上行 5下行
        default: disp_data=0;
        endcase
end
//显示译码器
always@(disp_data)
        begin
                case(disp_data)
                4'h1: seg=8'h06;//1楼
                4'h2: seg=8'h5b;//2楼
                4'h3: seg=8'h40;//待机
                4'h4: seg=8'h01;//上行
                4'h5: seg=8'h08;//下行
                default: seg=0;
                endcase
        end
endmodule

//6.蜂鸣==================================
module beep(
    input clk,
    input [1:0]state,
```

```
336     input [1:0]floor,
337     output reg beep
338     );
339     reg beep_n;//控制蜂鸣声时长
340     reg [24:0] beep_cnt=0;
341     reg [24:0] clk_500ms_cnt=0;
342     reg [24:0] clk_500ms_cnt2=0;
343       always @ (posedge clk)
344         begin
345             if(floor==1)
346               begin
347                 clk_500ms_cnt2=0;
348                 if (clk_500ms_cnt==24999999)//计数0.5s, 在此区间
                       发出蜂鸣声
349                     begin
350                         beep_n=0;
351                         clk_500ms_cnt=24999999;
352                     end
353                 else if(clk_500ms_cnt!==24999999&&clk_500ms_cnt
                       !==0)
354                     begin
355                         clk_500ms_cnt<=clk_500ms_cnt+1;
356                         beep_n=1;
357                     end
358                 else if(clk_500ms_cnt==0)
359                     begin
360                         clk_500ms_cnt<=clk_500ms_cnt+1;
361                         beep_n=0;
362                     end
363                 end
364             else
365                 begin
366                   clk_500ms_cnt=0;
367                 if (clk_500ms_cnt2==24999999)
```

```
368                 //计数0.5s，在此区间发出蜂鸣声
369                     begin
370                       beep_n=0;
371                       clk_500ms_cnt2=24999999;
372                     end
373                 else if(clk_500ms_cnt2!==24999999&&clk_500ms_cnt2
                       !==0)
374                      begin
375                         clk_500ms_cnt2<=clk_500ms_cnt2+1;
376                         beep_n=1;
377                      end
378                 else if(clk_500ms_cnt2==0)
379                      begin
380                         clk_500ms_cnt2<=clk_500ms_cnt2+1;
381                         beep_n=0;
382                      end
383                 end
384
385           if(beep_n==1)
386               begin
387                 if(beep_cnt==49000)
388                 //向蜂鸣器输入一个约500Hz的脉冲，使蜂鸣器发声
389                   begin
390                    beep=~beep;
391                    beep_cnt=0;
392                   end
393                 else
394                   begin beep_cnt=beep_cnt+1;end
395               end
396           else
397               begin
398                beep=0;
399                beep_cnt=0;
400               end
```

```
401     end
402 endmodule
403 
404 //约束文件=====================================
405 set_property IOSTANDARD LVCMOS33 [get_ports {dig[5]}]
406 set_property IOSTANDARD LVCMOS33 [get_ports {dig[4]}]
407 set_property IOSTANDARD LVCMOS33 [get_ports {dig[3]}]
408 set_property IOSTANDARD LVCMOS33 [get_ports {dig[2]}]
409 set_property IOSTANDARD LVCMOS33 [get_ports {dig[1]}]
410 set_property IOSTANDARD LVCMOS33 [get_ports {dig[0]}]
411 set_property IOSTANDARD LVCMOS33 [get_ports {led[3]}]
412 set_property IOSTANDARD LVCMOS33 [get_ports {led[2]}]
413 set_property IOSTANDARD LVCMOS33 [get_ports {led[1]}]
414 set_property IOSTANDARD LVCMOS33 [get_ports {led[0]}]
415 set_property IOSTANDARD LVCMOS33 [get_ports {seg[7]}]
416 set_property IOSTANDARD LVCMOS33 [get_ports {seg[6]}]
417 set_property IOSTANDARD LVCMOS33 [get_ports {seg[5]}]
418 set_property IOSTANDARD LVCMOS33 [get_ports {seg[4]}]
419 set_property IOSTANDARD LVCMOS33 [get_ports {seg[3]}]
420 set_property IOSTANDARD LVCMOS33 [get_ports {seg[2]}]
421 set_property IOSTANDARD LVCMOS33 [get_ports {seg[1]}]
422 set_property IOSTANDARD LVCMOS33 [get_ports {seg[0]}]
423 set_property IOSTANDARD LVCMOS33 [get_ports beep]
424 set_property IOSTANDARD LVCMOS33 [get_ports clk]
425 set_property PACKAGE_PIN T5 [get_ports {led[3]}]
426 set_property PACKAGE_PIN R7 [get_ports {led[2]}]
427 set_property PACKAGE_PIN R8 [get_ports {led[1]}]
428 set_property PACKAGE_PIN P9 [get_ports {led[0]}]
429 set_property PACKAGE_PIN D4 [get_ports clk]
430 set_property PACKAGE_PIN L2 [get_ports beep]
431 set_property PACKAGE_PIN N11 [get_ports {dig[5]}]
432 set_property PACKAGE_PIN N14 [get_ports {dig[4]}]
433 set_property PACKAGE_PIN N13 [get_ports {dig[3]}]
434 set_property PACKAGE_PIN M12 [get_ports {dig[2]}]
```

```
435 set_property PACKAGE_PIN H13 [get_ports {dig[1]}]
436 set_property PACKAGE_PIN G12 [get_ports {dig[0]}]
437 set_property PACKAGE_PIN L13 [get_ports {seg[7]}]
438 set_property PACKAGE_PIN M14 [get_ports {seg[6]}]
439 set_property PACKAGE_PIN P13 [get_ports {seg[5]}]
440 set_property PACKAGE_PIN K12 [get_ports {seg[4]}]
441 set_property IOSTANDARD LVCMOS33 [get_ports reset]
442 set_property IOSTANDARD LVCMOS33 [get_ports start]
443 set_property PACKAGE_PIN F3 [get_ports reset]
444 set_property PACKAGE_PIN T9 [get_ports start]
445 set_property PACKAGE_PIN K13 [get_ports {seg[3]}]
446 set_property PACKAGE_PIN L14 [get_ports {seg[2]}]
447 set_property PACKAGE_PIN N12 [get_ports {seg[1]}]
448 set_property PACKAGE_PIN P11 [get_ports {seg[0]}]
449 set_property PACKAGE_PIN K3 [get_ports row]
450 set_property IOSTANDARD LVCMOS33 [get_ports row]
451 set_property IOSTANDARD LVCMOS33 [get_ports key0]
452 set_property IOSTANDARD LVCMOS33 [get_ports key1]
453 set_property IOSTANDARD LVCMOS33 [get_ports key2]
454 set_property IOSTANDARD LVCMOS33 [get_ports key3]
455 set_property PACKAGE_PIN R12 [get_ports key0]
456 set_property PACKAGE_PIN T12 [get_ports key1]
457 set_property PACKAGE_PIN R11 [get_ports key2]
458 set_property PACKAGE_PIN T10 [get_ports key3]
459 set_property IOSTANDARD LVCMOS33 [get_ports {led_b[4]}]
460 set_property IOSTANDARD LVCMOS33 [get_ports {led_b[3]}]
461 set_property IOSTANDARD LVCMOS33 [get_ports {led_b[2]}]
462 set_property IOSTANDARD LVCMOS33 [get_ports {led_b[1]}]
463 set_property IOSTANDARD LVCMOS33 [get_ports {led_b[0]}]
464 set_property PACKAGE_PIN T2 [get_ports {led_b[0]}]
465 set_property PACKAGE_PIN R1 [get_ports {led_b[1]}]
466 set_property PACKAGE_PIN G5 [get_ports {led_b[2]}]
467 set_property PACKAGE_PIN H3 [get_ports {led_b[3]}]
468 set_property PACKAGE_PIN E3 [get_ports {led_b[4]}]
```