

6. 密码锁实验例程

6.1 实验目的

利用 MES22GP 板卡上的按键，拨码开关以及数码管实现一种简单的密码锁。

利用拨码开关设置密码，使用按键输入开锁密码。当开锁密码与设定密码相同时开锁成功，数码管显示 8888，密码错误时显示 7777。

SW1~SW8 设置 4 位数密码，每两位设置一位密码，SW[2:1]设置第一位，SW[4:3]设置第二位，SW[6:5]设置第三位，SW[8:7]设置第四位。四位密码最高可设置为“3333”；

KEY4~KEY1 作为密码输入，按键按一下数字加 1，数字由数码管显示，数字在 0，1，2，3 中循环。

K7 作为确认按键，按下 K7，输入的密码与设置的密码比对，如相同则显示 8888，若不同则显示 7777。

按下 K8 清零，按下后数码管显示 0000，可以重新输密码。

6.2 实验原理

原理上与前一个章节的序列检测是类似的，在前一个实验的基础上有了一些延伸；

序列对比的位宽发生改变，单个数据占 2bit，一个按键控制输入密码数据设置为 2bit 即可；对比与重新开始在此实验用两个按键实现，一个确认对比，一个清空结果；

6.3 实验源码设计

根据需求我们需要如下三个子模块：

①按键控制模块；

1、对 6 个按键输入信号均做消抖处理，2、KEY[4: 1]以下降沿来变更各自的输入密码，每次数字加 1（0~3 循环，2bit 即可）

②数码管显示模块；

显示状态有两种：

密码输入状态：

1、上电默认状态； 2、KEY8 触发进入重置状态； 3、实时显示 4 位输入密码；

密码验证状态：

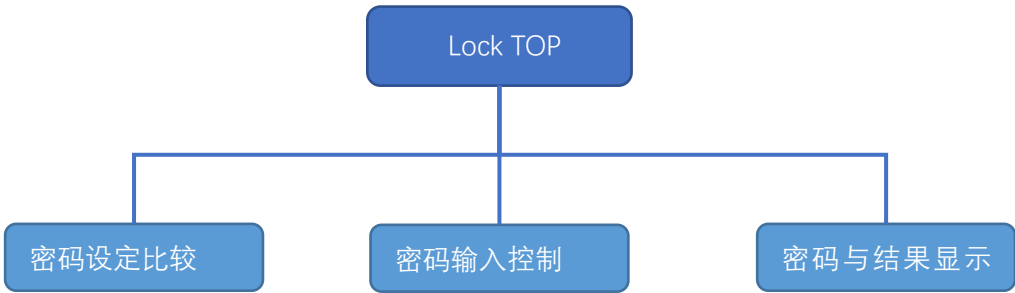
- 1、KEY7 触发进入；
- 2、显示密码验证结果，正确则显示 8888，错误则显示 7777；
- ③密码验证模块；

KEY7 下降沿触发使能工作； KEY7 下降沿触发所存输入密码，并与拨码开关设置的密码进行比较；

输出密码比较结果，提供个数码管显示模块。

6.3.1顶层模块设计

顶层模块与上述三个模块之间的关系如下图：



输入输出信号如下表：

信号	位宽	方向	描述
clk	1	输入	外部输入时钟，MES22GP 板卡输入时钟为 50MHz
key	4	输入	轻触按键输入信号，MES22GP 板卡上 K1~K4 输入
enter	1	输入	密码确认比对信号，MES22GP 板卡上 K7 输入
init	1	输入	密码确认重新输入信号，MES22GP 板卡上 K8 输入
sw	8	输入	密码设置输入信号，MES22GP 板卡上 SW1~8 输入
smg	8	输出	密码对比结果显示数码管段选信号输出
dig	4	输出	密码对比结果显示数码管位选信号输出

Module 设计如下：

```

1  `timescale 1ns / 1ps
2  `define UD #1
3  module lock_top(
4      input      clk,
5      input  [3:0] key,
6      input      enter,
7      input      init,
8      input  [7:0] sw,
9
10     output [7:0] smg,
11     output [3:0] dig
12 );
13
14 wire      enter_trig;
15 wire      init_trig;
16 wire [7:0] ctrl;
17 wire      com_result;
18
19 key_ctl key_ctl(
20     .clk      ( clk      ),//input      clk,
21     .key       ( key      ),//input  [3:0] key,
22     .enter     ( enter    ),//input      enter,
23     .init      ( init     ),//input      init,
24
25     .enter_trig ( enter_trig ),//output      enter_trig,
26     .init_trig  ( init_trig  ),//output      init_trig,
27     .ctrl       ( ctrl      ) //output  [7:0] ctrl
28 );
29
30 compare compare(
31     .clk      ( clk      ),//input      clk,
32     .sw       ( sw       ),//input [7:0] sw,
33     .ctrl     ( ctrl     ),//input [7:0] ctrl,
34     .enter_trig ( enter_trig ),//input      enter_trig,
35
36     .com_result ( com_result ) //output      com_result
37 );
38
39 seq_display(
40     .clk      ( clk      ),//input      clk,
41     .enter_trig ( enter_trig ),//input      enter_trig,
42     .init_trig  ( init_trig  ),//input      init_trig,
43     .com_result ( com_result ),//input      com_result,
44     .ctrl       ( ctrl      ),//input  [7:0] ctrl,
45
46     .smg       ( smg      ),//output reg [7:0] smg,
47     .dig       ( dig      ) //output reg [3:0] dig
48 );
49
50 endmodule
51

```

6.3.2按键控制设计

信号	位宽	方向	描述
clk	1	输入	外部输入时钟, MES22GP 板卡输入时钟为 50MHz
key	4	输入	轻触按键输入信号, MES22GP 板卡上 K1~K4 输入
enter	1	输入	密码确认比对信号, MES22GP 板卡上 K7 输入
init	1	输入	密码重新输入信号, MES22GP 板卡上 K8 输入
enter_trig	1	输出	确认对比触发脉冲信号输出
init_trig	1	输出	进入重新输入密码触发脉冲信号输出
ctrl	8	输出	输入的 4 个密码输出, 与设置密码的位定义一致

输入输出信号如下表:

Module 设计如下:

```

1  `timescale 1ns / 1ps
2  `define UD #1
3  module key_ctl(
4      input          clk,
5      input          [3:0] key,
6      input          enter,
7      input          init,
8
9      output          enter_trig,
10     output          init_trig,
11     output          [7:0] ctrl
12 );
13
14     wire [5:0] btn_deb;
15     // 按键消抖
16     btn_deb#(
17         .BTN_WIDTH ( 4'd6 ) //parameter BTN_WIDTH = 4'd8
18     ) U_btn_deb
19     (
20         .clk        ( clk ),//input          clk,
21         .btn_in      ( {enter,init,key} ),//input [BTN_WIDTH-1:0] btn_in,
22
23         .btn_deb     ( btn_deb ) //output reg [BTN_WIDTH-1:0] btn_deb
24     );
25
26     reg [1:0] key1_push_cnt=2'd0;
27     reg [1:0] key2_push_cnt=2'd0;
28     reg [1:0] key3_push_cnt=2'd0;
29     reg [1:0] key4_push_cnt=2'd0;
30
31     reg btn1_deb_1d,btn2_deb_1d,btn3_deb_1d,btn4_deb_1d;
32     reg enter_deb_1d,init_deb_1d;
33
34     //确认和重新输入密码触发脉冲信号获取
35     assign enter_trig = ~btn_deb[5] & enter_deb_1d;
36     assign init_trig = ~btn_deb[4] & init_deb_1d;
37
38     always @(posedge clk)
39     begin
40         btn1_deb_1d <= `UD btn_deb[0];
41         btn2_deb_1d <= `UD btn_deb[1];
42         btn3_deb_1d <= `UD btn_deb[2];
43         btn4_deb_1d <= `UD btn_deb[3];
44         init_deb_1d <= `UD btn_deb[4];
45         enter_deb_1d <= `UD btn_deb[5];
46     end
47

```

```

48     always @(posedge clk)
49     begin
50         if(~btn_deb[4] & init_deb_1d)
51             key1_push_cnt <= `UD 2'd0;
52         else if(~btn_deb[0] & btn1_deb_1d)
53             begin
54                 key1_push_cnt <= `UD key1_push_cnt + 2'd1;
55             end
56     end
57
58     always @(posedge clk)
59     begin
60         if(~btn_deb[4] & init_deb_1d)
61             key2_push_cnt <= `UD 2'd0;
62         else if(~btn_deb[1] & btn2_deb_1d)
63             begin
64                 key2_push_cnt <= `UD key2_push_cnt + 2'd1;
65             end
66     end
67
68     always @(posedge clk)
69     begin
70         if(~btn_deb[4] & init_deb_1d)
71             key3_push_cnt <= `UD 2'd0;
72         else if(~btn_deb[2] & btn3_deb_1d)
73             begin
74                 key3_push_cnt <= `UD key3_push_cnt + 2'd1;
75             end
76     end
77
78     always @(posedge clk)
79     begin
80         if(~btn_deb[4] & init_deb_1d)
81             key4_push_cnt <= `UD 2'd0;
82         else if(~btn_deb[3] & btn4_deb_1d)
83             begin
84                 key4_push_cnt <= `UD key4_push_cnt + 2'd1;
85             end
86     end
87     //密码输出: {第四位密码, 第三位密码, 第二位密码, 第一位密码}, 每位密码占 2bit
88     assign ctrl1 = {key4_push_cnt, key3_push_cnt, key2_push_cnt, key1_push_cnt};
89
90 endmodule
91

```

6.3.3对比模块设计

输入输出信号如下表:

信号	位宽	方向	描述
clk	1	输入	外部输入时钟, MES22GP 板卡输入时钟为 50MHz
sw	8	输入	设置 4 位密码
enter_trig	1	输入	确认对比触发脉冲信号输入

ctrl	8	输入	输入的 4 位密码,
com_result	1	输出	输出对比结果

Module 设计

```

1  `timescale 1ns / 1ps
2  `define UD #1
3  module compare(
4      input      clk,
5      input [7:0] sw,
6      input [7:0] ctrl,
7      input      enter_trig,
8
9      output      com_result
10 );
11 //=====
12 //锁存当前的输入密码;
13 reg [7:0] ctrl_1d;
14 always @(posedge clk)
15 begin
16     if(enter_trig)
17         ctrl_1d <= `UD ctrl;
18     end
19
20     assign com_result = (ctrl_1d == sw);
21 endmodule

```

6.3.4显示模块设计

输入输出信号如下表:

信号	位宽	方向	描述
clk	1	输入	外部输入时钟, MES22GP 板卡输入时钟为 50MHz
enter_trig	1	输入	确认对比触发脉冲信号输入
init_trig	1	输入	重新输入密码触发脉冲信号输入
com_result	1	输入	输入比对模块的对比结果
ctrl	8	输入	输入的 4 位密码,
smg	8	输出	数码管段选输出
dig	4	输出	数码管位选输出

此模块设计需要注意数码管显示的两种模式: 密码输入模式与密码对比结果显示模式; 两种模式的切换由 enter_trig 与 init_trig 触发进入;

对于数码管的显示控制模块这里就不重复描述了;

```

1  `timescale 1ns / 1ps
2  `define UD #1
3  module seq_display(
4      input          clk,
5      input          enter_trig,
6      input          init_trig,
7      input          com_result,
8      input          [7:0]  ctrl,
9
10     output reg [7:0]  smg,
11     output reg [3:0]  dig
12 );
13
14     //=====
15     //显示状态区分
16     reg      seq_status= 1'b0;
17     always @(posedge clk)
18     begin
19         if(enter_trig)
20             seq_status <= `UD 1'b1;
21         else if(init_trig)
22             seq_status <= `UD 1'b0;
23     end
24     //=====
25     //数码管显示控制
26     reg [3:0] key0_cnt=4'd0,key1_cnt=4'd0,key2_cnt=4'd0,key3_cnt=4'd0;
27     always @(posedge clk)
28     begin
29         if(seq_status)
30         begin
31             if(com_result)
32             begin
33                 key0_cnt <= `UD 4'd8;
34                 key1_cnt <= `UD 4'd8;
35                 key2_cnt <= `UD 4'd8;
36                 key3_cnt <= `UD 4'd8;
37             end
38             else
39             begin
40                 key0_cnt <= `UD 4'd7;
41                 key1_cnt <= `UD 4'd7;
42                 key2_cnt <= `UD 4'd7;
43                 key3_cnt <= `UD 4'd7;
44             end
45         end
46         else
47         begin
48             key0_cnt <= `UD {2'd0,ctrl[1:0]};
49             key1_cnt <= `UD {2'd0,ctrl[3:2]};
50             key2_cnt <= `UD {2'd0,ctrl[5:4]};
51             key3_cnt <= `UD {2'd0,ctrl[7:6]};
52         end
53     end
54 
```

```
55 //=====
56 //          时钟分频
57 wire clk_100khz;
58 div_clk div_clk (
59     .clk      (clk),
60     .clk_100khz (clk_100khz)
61 );
62 //=====
63 //          数码管显示
64 reg [1:0]sel=0;
65 always @(posedge clk_100khz)
66 begin
67     sel <= `UD sel+1'b1;
68 end
69
70 wire [3:0]dig0;
71 wire [7:0]smg0;
72 seq_control seq_control_0
73 (
74     .sel(2'd3),
75     .key(key0_cnt),
76     .dig(dig0),
77     .smg(smg0)
78 );
79
80 wire [3:0]dig1;
81 wire [7:0]smg1;
82 seq_control seq_control_1
83 (
84     .sel(2'd2),
85     .key(key1_cnt),
86     .dig(dig1),
87     .smg(smg1)
88 );
89
90 wire [3:0]dig2;
91 wire [7:0]smg2;
92 seq_control seq_control_2
93 (
94     .sel(2'd1),
95     .key(key2_cnt),
96     .dig(dig2),
97     .smg(smg2)
98 );
99
100 wire [3:0]dig3;
101 wire [7:0]smg3;
102 seq_control seq_control_3
103 (
104     .sel(2'd0),
105     .key(key3_cnt),
106     .dig(dig3),
107     .smg(smg3)
108 );
109
```

```
110 always @(posedge clk_100khz)
111 begin
112     if(sel==2'b00)
113         dig <= `UD dig0;
114     else if(sel==2'b01)
115         dig <= `UD dig1;
116     else if(sel==2'b10)
117         dig <= `UD dig2;
118     else if(sel==2'b11)
119         dig <= `UD dig3;
120 end
121
122 always @(posedge clk_100khz)
123 begin
124     if(sel==2'b00)
125         smg <= `UD smg0;
126     else if(sel==2'b01)
127         smg <= `UD smg1;
128     else if(sel==2'b10)
129         smg <= `UD smg2;
130     else if(sel==2'b11)
131         smg <= `UD smg3;
132 end
133
134 endmodule
135
```

6.4 实验现象

实验步骤:

- 1、调整输入序列，更改拨码开关的输入值（SW[8: 1]）；
- 2、调整固定序列，通过轻触按键调整输入密码，数码管实时显示输入密码；
- 3、按下轻触按键 KEY7，触发进行密码比对，并且数码管显示比对结果；
- 4、按下轻触按键 KEY8，进入重新输入密码状态，重新执行前面三个步骤。

实验现象:

当 SW[8:1]=8'b10101010;当输入密码状态时显示 4 个 2，按下 Key7 后数码管显示数字 8888;当输入密码状态时显示不是 4 个 2 时 2，按下 Key7 后数码管显示数字 7777; 按下 Key8 后重新调整密码，进入输入密码状态。