串行密码锁实验报告

翁家翌 2016011446

2018.5.18

1 实验目的

1. 学习使用状态机控制电路工作，在不同状态下完成相应的功能。

2. 进一步掌握时序逻辑电路的基本分析和设计方法。

3. 学会利用仿真软件实现对数字电路的逻辑功能进行验证和分析。

2 实验内容

1. 设计一个 4 位 16 进制串行密码锁，其具体功能如下：

设置密码 用户可串行设置 4 位 16 进制密码。

验证密码 用户串行输入密码，如果密码符合则点亮开锁灯，若不符合则点亮错误灯。

2. 研究内容：

密码预置 为管理员创建万用密码以备管理。

系统报警 开锁 3 次失败后点亮报警灯，并锁定密码锁，只有输入管理员密码才可开锁，并解除报警。

3 代码及注释

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_arith.all;
4  use ieee.std_logic_unsigned.all;
5
6  entity lock is
7          port(
8                  code: in std_logic_vector(3 downto 0);
9                  mode: in std_logic_vector(1 downto 0);
10                 clk, rst: in std_logic;
11                 unlock: out std_logic;
12                 alarm, err: buffer std_logic
13         );
14         type passwd is array (3 downto 0) of integer;
15 end lock;
16
17 architecture arc of lock is
```

```
18          signal pwd: passwd;
19          signal state: integer := 0;
20          signal cnt: integer := 0; -- which bit (admin) or failure time (users)
21  begin
22          process(clk, rst)
23          begin
24                  if (rst = '0') then
25                          unlock <= '0';
26                          err <= '0';
27                          state <= 0;
28                          if (alarm = '1') then
29                                  cnt <= 0;
30                          end if;
31                  elsif (clk'event and clk = '0') then
32                          if (alarm = '1') then
33                                  if (CONV_INTEGER(code) = 8) then -- enter admin passwd 8888
34                                          if (cnt > 2) then
35                                                  cnt <= 0;
36                                                  alarm <= '0';
37                                                  state <= 0;
38                                          else
39                                                  cnt <= cnt + 1;
40                                          end if;
41                                  else
42                                          cnt <= 0;
43                                  end if;
44                          elsif (mode = "00" and err = '0') then -- set passwd
45                                  case state is
46                                          when 0 => pwd(0) <= CONV_INTEGER(code); state <= 1;
47                                          when 1 => pwd(1) <= CONV_INTEGER(code); state <= 2;
48                                          when 2 => pwd(2) <= CONV_INTEGER(code); state <= 3;
49                                          when 3 => pwd(3) <= CONV_INTEGER(code); state <= 7;
                                                unlock <= '1';
50                                          when others => NULL;
51                                  end case;
52                          elsif (mode = "01") then -- check passwd
53                                  case state is
54                                          when 0 =>
55                                                  if (CONV_INTEGER(code) = pwd(0)) then
56                                                          state <= 4;
57                                                          err <= '0';
58                                                  else
59                                                          err <= '1';
60                                                          if (cnt > 1) then
61                                                                  alarm <= '1';
62                                                                  cnt <= 0;
63                                                          else
64                                                                  cnt <= cnt + 1;
65                                                          end if;
66                                                  end if;
67                                          when 4 =>
68                                                  if (CONV_INTEGER(code) = pwd(1)) then
69                                                          state <= 5;
70                                                  else
71                                                          err <= '1';
72                                                          state <= 0;
73                                                          if (cnt > 1) then
74                                                                  alarm <= '1';
75                                                                  cnt <= 0;
```

```
76                                              else
77                                                      cnt <= cnt + 1;
78                                              end if;
79                                      end if;
80                              when 5 =>
81                                      if (CONV_INTEGER(code) = pwd(2)) then
82                                              state <= 6;
83                                      else
84                                              err <= '1';
85                                              state <= 0;
86                                              if (cnt > 1) then
87                                                      alarm <= '1';
88                                                      cnt <= 0;
89                                              else
90                                                      cnt <= cnt + 1;
91                                              end if;
92                                      end if;
93                              when 6 =>
94                                      if (CONV_INTEGER(code) = pwd(3)) then
95                                              state <= 7;
96                                              unlock <= '1';
97                                              cnt <= 0;
98                                      else
99                                              err <= '1';
100                                             state <= 0;
101                                             if (cnt > 1) then
102                                                     alarm <= '1';
103                                                     cnt <= 0;
104                                             else
105                                                     cnt <= cnt + 1;
106                                             end if;
107                                     end if;
108                             when others => NULL;
109                     end case;
110             end if;
111     end if;
112 end process;
113 end arc;
```

工作原理：采用状态机的方法，考虑各种情况，构造相应的状态转移图。我多设置了几个状态，使得能够在一个程序中完成了基本要求和提高要求两部分的内容，以下为具体每个状态的说明：(遇到时钟下降沿的时候进行判断)

0 号状态：初始状态，如果 mode 是 00，得到第一位密码，进入 1 号状态；如果 mode 是 01，判断第一位密码是否正确，正确则进入 4 号状态，错误则留在 0 号状态。0 号状态同时还是三次验证密码失败时的死状态，此时除非正确输入管理员密码才可以解除警报，否则不论什么 mode，都留在 0 号状态不操作。

1 号状态：如果 mode 是 00，得到第二为密码，进入 2 号状态，否则不操作。

2 号状态：如果 mode 是 00，得到第三位密码，进入 3 号状态，否则不操作。

3 号状态：如果 mode 是 00，得到第四位密码，进入 7 号状态，否则不操作。

4 号状态，如果 mode 是 01，判断第二位密码是否正确，正确的话进入 5 号状态，错误的话回到 0 号状态，否则不操作。

5 号状态，如果 mode 是 01，判断第三位密码是否正确，正确的话进入 6 号状态，错误的话回到 0 号状态，否则不操作。

6 号状态，如果 mode 是 01，判断第四位密码是否正确，正确的话进入 7 号状态，错

误的话回到 0 号状态，否则不操作。

7 号状态：成功设置/验证密码的死状态，除非遇到 rst 清零信号为 0，进入 0 号状态，否则皆不操作。

1-7 号状态，遇到 rst 为 0 的情况均回到初始状态，即 0 号状态，但是验证密码错误次数不清零。

4 仿真结果

如图1所示。

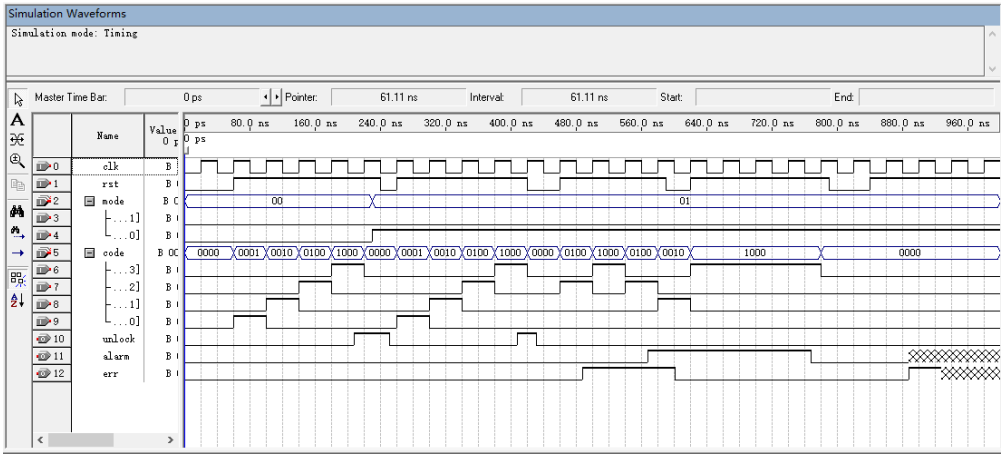

图 1: 仿真结果，包含设置密码，验证密码正确、三次错误触发警报、管理员解锁的效果

仿真结果说明：一开始清零，设置密码为 1248。清零验证密码，输入 1248 成功，unlock=1，对应二极管发光；接着失败三次，通过输入管理员密码 8888，alarm 警报消失。

5 实验小结

这是我第四次，也是本学期最后一次进行 CPLD 实验。在之前的实验的基础上又有了不小的提高，上网学习了 CONV_INTEGER 的命令，能够将表示二进制数组的 std_logic_vector 转化为整数 integer，同时这次代码中用上了 buffer、signal、clk'event、case 等各种命令，算是对之前所学的比较完整的总结与集体应用。

通过本次实验，熟悉了状态机的理论，知道如何通过 VHDL 实现状态机，并解决具体问题，也了解到状态机能够便于我们理解与编程的优点。除此之外，在完成此次代码的时候，用了比之前任意一次都少的时间，就完成了代码编写、仿真、导入、验证实验结果的完整过程，因为更加轻车熟路，反而有一种实验越来越简单的感觉。