

串行密码锁实验

赵晨阳 计 06

2020012363

实验目的

1. 学习使用状态机来控制电路工作，在不同的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. 学会利用软件仿真实现数字电路的逻辑功能的验证和分析。

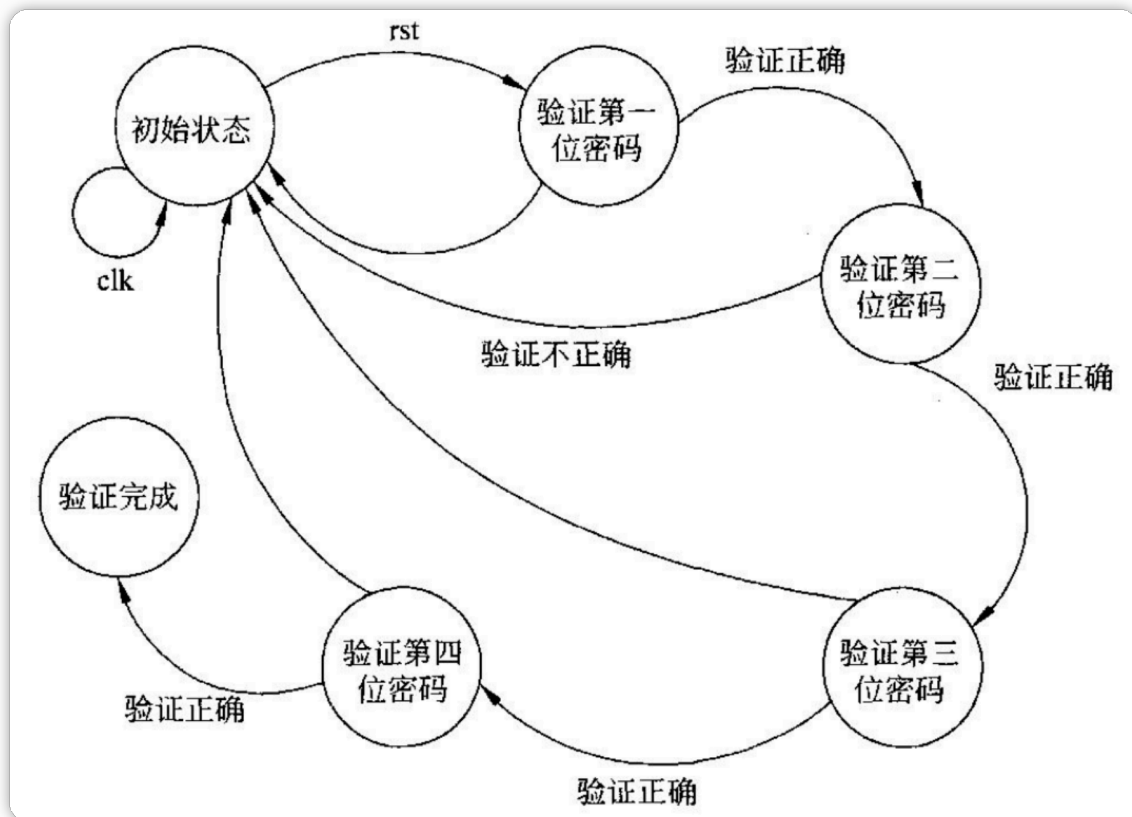
实验内容

使用状态机设计一个串行密码锁，包括如下内容：

功能	功能描述
预设密码	创建管理员万能密码（4位十六进制）以备管理，它在任何时候都能开锁
设置密码	用户可以串行设置4位十六进制密码
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
系统报警	连续开锁三次失败后点亮报警灯，此时锁定设置密码与用户密码验证功能，只有输入管理员密码才可以解锁

实验设计

状态设计



参考教科书上的实验设计，状态机一共 8 个状态，含义如下：

1. 状态 0：起始状态，当mode为“00”时设置第一位密码，并切换至状态1；当mode为“01”时验证第一位密码，若正确则切换至状态4
2. 状态1-3：设置密码状态，此时mode为“00”，依次设置第2-4位密码，设置完后切换至下一个状态，而状态3切换至状态7
3. 状态 4-6：验证密码状态，此时mode为“01”，依次验证第2-4位密码，验证成功后切换至下一个状态，而状态6切换至状态7；若验证失败，则记录失败次数，并将状态切换至状态0，当次数为3时，使警报灯亮
4. 状态 7：结束状态，此时锁被打开

具体实现

```

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         err , alarm : BUFFER STD_LOGIC
13     );
14     SUBTYPE pwd1 is INTEGER RANGE 0 TO 15;
15     TYPE password IS ARRAY (3 DOWNTO 0) OF pwd1;
16     SUBTYPE states is INTEGER RANGE 0 TO 7;
17 END lock;
18
19 ARCHITECTURE bhv OF lock IS
20     CONSTANT admin_password : password := (1, 3, 1, 4);
21     SIGNAL cur_password : password := admin_password;
22     SIGNAL input_password : pwd1;
23     SIGNAL s1, s2 : STD_LOGIC;
24     SIGNAL state : states;
25     SIGNAL count : INTEGER RANGE 0 TO 2 := 0;
26
27 BEGIN
28     input_password <= conv_integer(code);
29
30     PROCESS (clk, rst)
31     BEGIN
32         IF (rst = '1') THEN
33             unlock <= '0';
34             err <= '0';
35             state <= 0;
36
37         ELSIF (clk'event AND clk = '1') THEN
38             IF (mode = "00" AND alarm = '0' AND count = 0) THEN
39                 CASE state IS
40                     WHEN 0 | 1 | 2 =>
41                         cur_password(state) <= input_password;
42                         state <= state + 1;
43                     WHEN 3 =>
44                         cur_password(3) <= input_password;
45                         state <= 7;

```

```

46         unlock <= '1';
47         WHEN OTHERS => NULL;
48     END CASE;
49
50     ELSIF (mode = "01") THEN
51         CASE state IS
52             WHEN 0 =>
53                 IF ((input_password = cur_password(0) AND alarm =
54 '0')
55
56                     OR input_password = admin_password(0)) THEN
57                     IF (input_password = cur_password(0) AND alarm =
58 '0') THEN
59
60                         s2 <= '1';
61                     ELSE
62                         s2 <= '0';
63                     END IF;
64
65                     IF (input_password = admin_password(0)) THEN
66                         s1 <= '1';
67                     ELSE
68                         s1 <= '0';
69                     END IF;
70
71                     state <= 4;
72                     err <= '0';
73
74                     ELSE
75                         err <= '1';
76                         IF (count > 1) THEN
77                             alarm <= '1';
78                             count <= 0;
79                         ELSE
80                             count <= count + 1;
81                         END IF;
82
83                     END IF;
84
85                 WHEN 4 | 5 | 6 =>

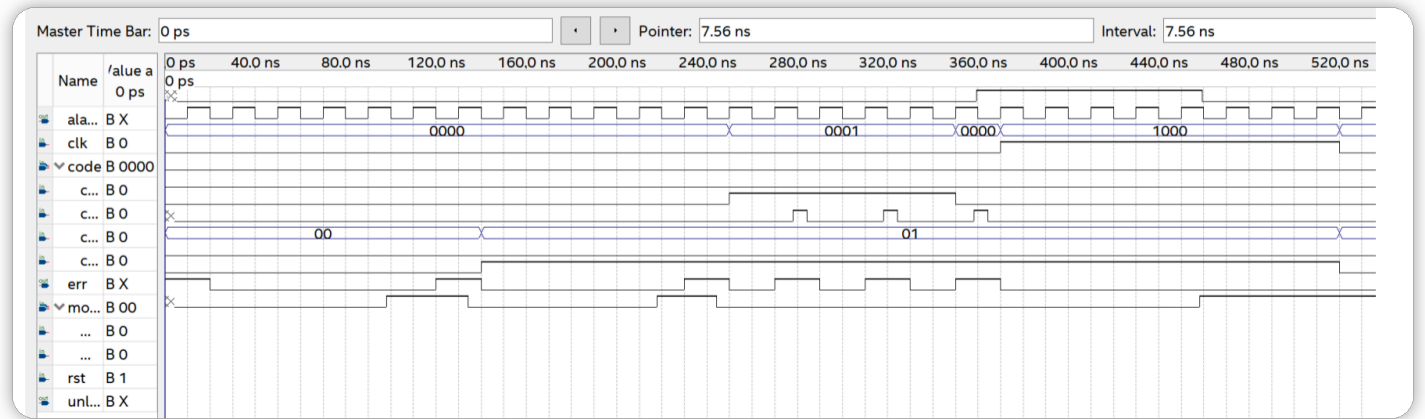
```

```

82         IF ((input_password = cur_password(state-3) AND (s2
= '1')) OR (input_password = admin_password(state-3) AND (s1 =
'1')))) THEN
83             IF (state = 6) THEN
84                 unlock <= '1';
85                 alarm <= '0';
86                 count <= 0;
87             END IF;
88
89             state <= state + 1;
90
91             IF (input_password /= cur_password(state-3)) THEN
92                 s2 <= '0';
93             END IF;
94             IF (input_password /= admin_password(state-3))
THEN
95                 s1 <= '0';
96             END IF;
97
98             ELSE
99                 err <= '1';
100                state <= 0;
101                IF (count > 1) THEN
102                    alarm <= '1';
103                    count <= 0;
104                ELSE
105                    count <= count + 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 bhv;

```

实验仿真



首先，我们将密码设置为0000，然后用0000验证密码，解锁；然后用0001验证密码，重复3次，进入警报状态，最后用管理员密码1000解开警报。