串行密码锁实验报告

计 71 张程远 2017011429

1、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，如果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，只有用管理员密码才可开锁，并解除报警。

3、实验代码

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity lock is
    port(
        code: in std_logic_vector(3 downto 0);
        mode: in std_logic_vector(1 downto 0);
        clk, rst: in std_logic;
        unlock: out std_logic;
        alarm, err: buffer std_logic
    );
    type passwd is array (3 downto 0) of integer;
end lock;

architecture arc of lock is
    signal pwd: passwd;
    signal sp: integer := 0; --验证password
```

```
    signal sg: integer := 0; --验证管理员密码
    signal state: integer := 0; --表示状态
    signal cnt: integer := 0;
begin
    process(clk, rst)
    begin
        if (rst = '0') then
            unlock <= '0';
            err <= '0';
            state <= 0;
            sp <= 0;
            sg <= 0;
            if (alarm = '1') then
                cnt <= 0;
            end if;
        elsif (clk'event and clk = '1') then
            if (alarm = '1') then
                if (CONV_INTEGER(code) = 8) then -管理员密码为8888
                    if (cnt > 2) then
                        cnt <= 0;
                        alarm <= '0';
                        state <= 0;
                        err <= '0';
                        unlock <= '1';
                    else
                        cnt <= cnt + 1;
                    end if;
                else
                    cnt <= 0;
                end if;
            elsif (mode = "00" and err = '0') then --“00“状态设置密码
                case state is
                    when 0 => pwd(0) <= CONV_INTEGER(code); state <= 1;
                    when 1 => pwd(1) <= CONV_INTEGER(code); state <= 2;
                    when 2 => pwd(2) <= CONV_INTEGER(code); state <= 3;
                    when 3 => pwd(3) <= CONV_INTEGER(code); state <= 7;
unlock <= '1';
                    when others => NULL;
                end case; --设置密码结束
            elsif (mode = "01") then -验证密码状态
                case state is
                    when 0 =>
                        if(CONV_INTEGER(code) /= pwd(0)) then
                            sp <= 1; --sp等于1则password匹配不正确
```

```
                        end if;
                        if(CONV_INTEGER(code) /= 8) then
                            sg <= 1; --sg等于1则管理员密码匹配不正确
                        end if;
                        if ((CONV_INTEGER(code) = pwd(0) and sp = 0) or
(CONV_INTEGER(code) = 8 and sg = 0)) then
                            state <= 4; --状态转移
                            err <= '0';
                        else
                            err <= '1';
                            if (cnt > 1) then
                                alarm <= '1';
                                cnt <= 0;
                            else
                                cnt <= cnt + 1;
                            end if;
                        end if;
                    when 4 =>
                        if(CONV_INTEGER(code) /= pwd(1)) then
                            sp <= 1;
                        end if;
                        if(CONV_INTEGER(code) /= 8) then
                            sg <= 1;
                        end if;
                        if ((CONV_INTEGER(code) = pwd(1) and sp = 0) or
(CONV_INTEGER(code) = 8 and sg = 0)) then
                            state <= 5;
                        else
                            err <= '1';
                            state <= 0;
                            if (cnt > 1) then
                                alarm <= '1';
                                cnt <= 0;
                            else
                                cnt <= cnt + 1;
                            end if;
                        end if;
                    when 5 =>
                        if(CONV_INTEGER(code) /= pwd(2)) then
                            sp <= 1;
                        end if;
                        if(CONV_INTEGER(code) /= 8) then
                            sg <= 1;
                        end if;
```

```
                        if ((CONV_INTEGER(code) = pwd(2) and sp = 0) or
(CONV_INTEGER(code) = 8 and sg = 0)) then
                            state <= 6;
                        else
                            err <= '1';
                            state <= 0;
                            if (cnt > 1) then
                                alarm <= '1';
                                cnt <= 0;
                            else
                                cnt <= cnt + 1;
                            end if;
                        end if;
                    when 6 =>
                        if(CONV_INTEGER(code) /= pwd(3)) then
                            sp <= 1;
                        end if;
                        if(CONV_INTEGER(code) /= 8) then
                            sg <= 1;
                        end if;
                        if ((CONV_INTEGER(code) = pwd(3) and sp = 0) or
(CONV_INTEGER(code) = 8 and sg = 0)) then
                            state <= 7;
                            unlock <= '1'; --开锁
                            cnt <= 0;
                        else
                            err <= '1';
                            state <= 0;
                            if (cnt > 1) then
                                alarm <= '1';
                                cnt <= 0;
                            else
                                cnt <= cnt + 1;
                            end if;
                        end if;
                    when others => NULL;
                end case;
            end if;
        end if;
    end process;
end arc;
```

工作原理：采用状态机的方法，通过构造状态转移，使得其接受正确的输入，并让错误密码一直待在 0 号状态。以下为具体每个状态的解释：

0 号状态：初始状态，考虑 mode 的值。当 mode=00 时，进入设置密码状态，用户得到第一位密码进入 1 号状态；当 mode=01 时，验证第一位密码是否有效，如果有效（等于用户密码或管理员密码第一位）则进入状态 4，否则一直停留在状态 0。

1、2 状态：中间状态。在这两个状态，用户分别设置第 2、3 位密码，进入 3 号状态。

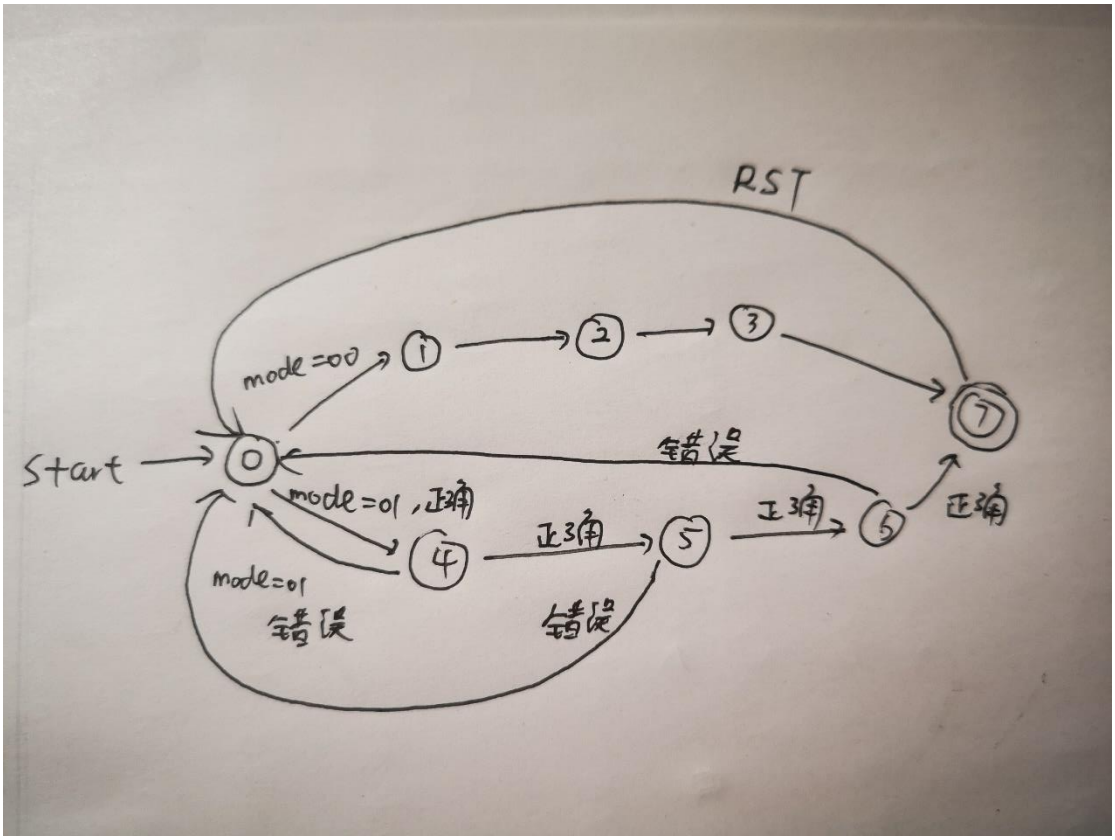
3 号状态：用户设置第 4 位密码进入 7 号状态，提示灯亮起，设置完毕，需要 reset 重新进入 00 模式或 01 模式。

4、5 状态：中间状态。用户验证第 2、3 位密码，如果仍然正确进入 6 号状态，否则立即返回 0 号状态。

6 号状态：用户验证第 4 位密码，正确则进入 7 号状态，否则返回 0 号状态。

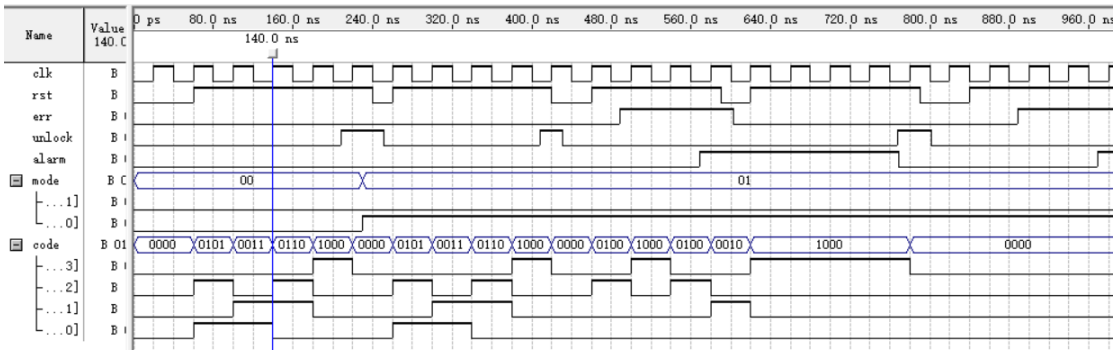
7 号状态：开锁状态，用户需按 reset 进行新的操作。

状态转移图如下：



同时有一个 cnt 记录错误次数。如果达到 3 则 alarm 亮起，之后的输入都将与管理员密码比对，如果错误就一直停留在当前情况，直到密码正确。

4、仿真结果



设置初始密码 5368，仿真图提供了输入正确密码、输入 3 次错误密码以及输入管理员密码的仿真波形情况。

5、实验小结

本次实验的要点在于状态机的使用。关于状态机的设计，我在自动机课程上已经有所了解，所以设计出一个使用状态机的密码锁并非是一件难事。不过编程实现时要考虑很多细节，如信号归零、错误次数归零、如何判断与管理员密码相同前缀且与用户密码有相同后缀的错误密码等，都是需要考虑的问题。

本次实验在演示的时候并不顺利，因为 CLK 按键具有严重的毛刺现象，摁一下有时相当于摁了两下，这样就会造成错误。我前后演示了 3 次，将功能分解拆开，最后才完整展示出了整个密码锁的功能。感谢老师和助教的理解和指导！这是最后一次 CPLD 实验了，希望我在接下来的实验测试中能够顺利完成。