

串行密码锁实验报告

项晨东 2019011831

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 实验代码以及注释

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

entity password 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 password;

architecture arc of password is
    constant admin: passwd := (9,9,9,9); -- admin password
    signal pwd: passwd := admin;
    signal state: integer :=0;
    signal cnt: integer := 0;
```

```

signal input: integer := 0;
signal is_admin, user: integer:=0; -- check whether the user or admin input
begin
process(clk, rst)
begin
    input <= conv_integer(code);
    if(rst = '1') then -- reset
        unlock <= '0';
        err <= '0';
        state <= 0;
    elsif(clk'event and clk = '1')then
        if(mode = "00" and alarm = '0' and cnt = 0) then --set mode
            case state is
                when 0|1|2 => pwd(state) <= input; state <= state +1;
                when 3 => pwd(state)<= input; state <= 7; unlock <= '1';
                when others => NULL;
            end case;
        elsif (mode = "01") then --checking password
            case state is
                when 0 =>
                    if((input = pwd(0)) and (alarm = '0')) then -- judge password type
                        user <= 1;
                    else
                        user <= 0;
                    end if;
                    if(input = admin(0)) then
                        is_admin <= 1;
                    else
                        is_admin <= 0;
                    end if;
                    -- move state
                    if(((input = pwd(0)) and (alarm = '0')) or(input = admin(0))) then
                        state <= 4;
                        err <= '0';
                    else
                        err <= '1';
                        cnt <= cnt +1;
                        if (cnt > 1) then
                            alarm <= '1';
                            cnt <= 0;
                        end if;
                    end if;
                when 4|5|6 => -- continue move state
                    if((pwd(state-3)=input and (user=1))or
                        (admin(state-3)=input and is_admin=1))then
                        if(state = 6) then -- unlocked
                            unlock <= '1';
                            alarm <= '0';
                            cnt <= 0;

```

```

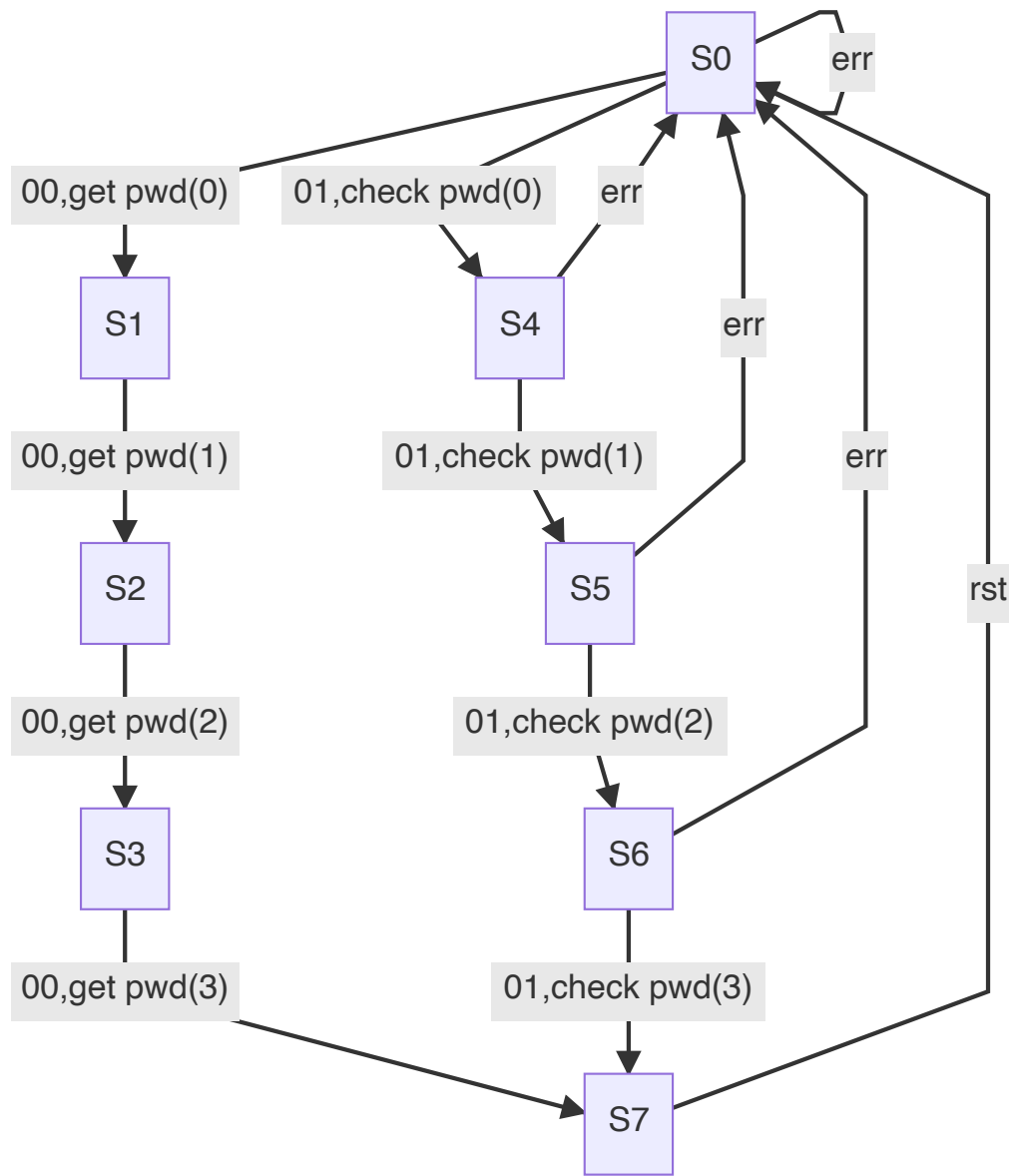
        end if;
        state <= state +1;
        if(pwd(state-3)/=input)then -- judge password type
            user <= 0;
        end if;
        if(admin(state-3)/=input)then
            is_admin <=0;
        end if;
    else -- error processing
        err<='1';
        state <= 0;
        cnt <= cnt +1;
        if (cnt > 1)then -- too many fail time
            cnt <= 0;
            alarm <= '1';
        end if;
    end if;
    when others => NULL;
end case;
end if; -- mode level
end if; -- event level
end process;
end arc;

```

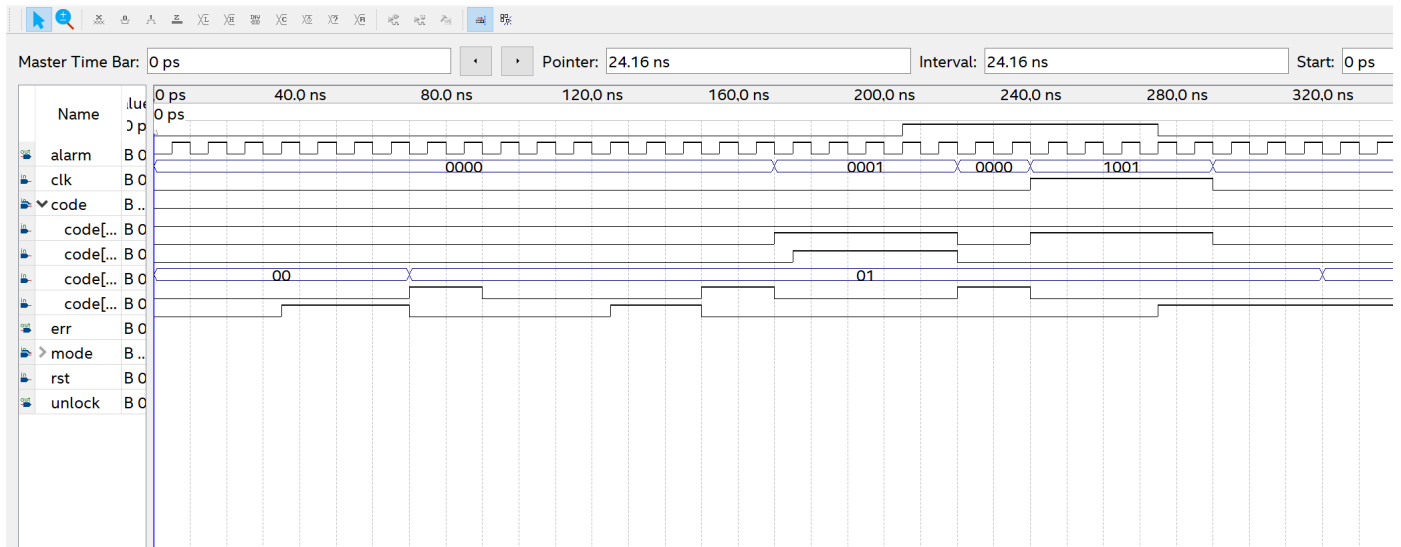
4 工作原理

状态机工作原理如下, 其中 `sx` 表示各个状态, 边上的 `00` 以及 `01` 表示模式。`err` 表示校验不正确。

任意状态下, 按下 `rst` 键均可返回到 `s0`。对于 `alarm` 的处理和在 `01` 模式下, 若 `alarm` 为1, 则只有管理员密码能够通过校验。



5 仿真结果



仿真说明:

1. 一开始设置密码为 0000 , 后进行 rst 。
2. 然后进入 01 模式验证 0000 , 密码开锁, 后进行 rst 。
3. 用 1111 进行错误密码尝试, 拉响警报, 后进行 rst 。
4. 用管理员密码 9999 解除警报。

6 实验小结

这次实验让我对 vhd1 的代码更加熟悉, 学会使用 conv_integer 来将逻辑阵列转化为数字, 学会了状态机的设计和实现。同时debug的能力也更强, 更熟悉语言特性, 纠正了一些习惯性的错误(例如赋值要用<=, 习惯写成=)。总体来说实验的过程更加熟练, 并且有所收获。

非常感谢老师和助教的辛苦付出~