

《数字逻辑》实验报告

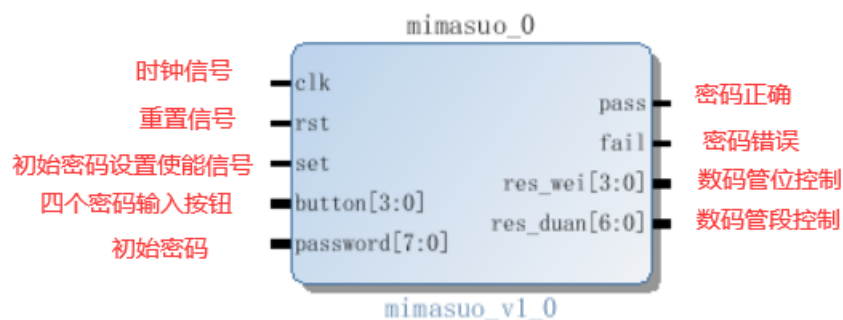
姓名	李燕琴, 杨思怡	年级	19 级
学号	20195633, 20195217	专业、班级	计算机科学与技术卓越 02/01 班
实验名称	项目三 密码锁设计		
实验时间	2020 年 12 月 2 日	实验地点	DS1410
实验成绩		实验性质	<input type="checkbox"/> 验证性 <input type="checkbox"/> 设计性 <input type="checkbox"/> 综合性
<p>教师评价：</p> <p><input type="checkbox"/>算法/实验过程正确； <input type="checkbox"/>源程序/实验内容提交 <input type="checkbox"/>程序结构/实验步骤合理；</p> <p><input type="checkbox"/>实验结果正确； <input type="checkbox"/>语法、语义正确； <input type="checkbox"/>报告规范；</p> <p>评语：</p> <p>评价教师签名（电子签名）：</p>			
<p>一、实验目的</p> <p>通过实验，巩固有限状态机设计方法，并能实现密码锁设计。</p>			
<p>二、实验项目内容</p> <p>1、设计一个密码锁电路，可以通过拨码开关设置初始密码，通过按钮来输入</p> <p>密码，密码位数至少为 4 位，判断输入密码与设置的密码是否一致。</p> <p>2、画出密码锁电路的状态转换图。</p> <p>3、编写密码锁模块并仿真。</p> <p>4、编写顶层模块，综合、实验、生成 bit 流，下载到 Neys4 开发板</p>			

进行验证。

三、实验设计

总体设计思想：采用分模块化设计思想，主要分为顶层模块，数据输入模块，密码检测状态机模块，数码管显示模块，按键防抖模块，分频模块。

(1) 顶层模块设计



其中 `rst` 信号将初始设置的密码重置。

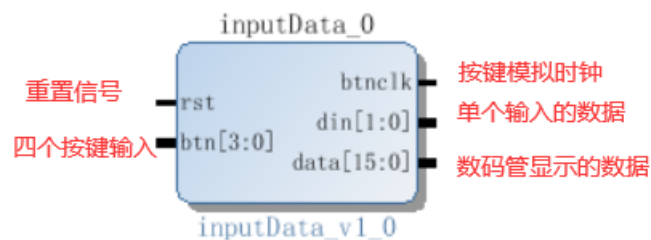
主要内部信号：

内置按键模拟时钟：`btnclk`；

单次输入密码数据：`[1:0]din`;

数码管显示数据：`[15:0] data`;

(2) 数据输入模块设计



其中 `rst` 信号将 `din` 重置。

按下的按键(`btn`)与数据(`din`)的对应关系：

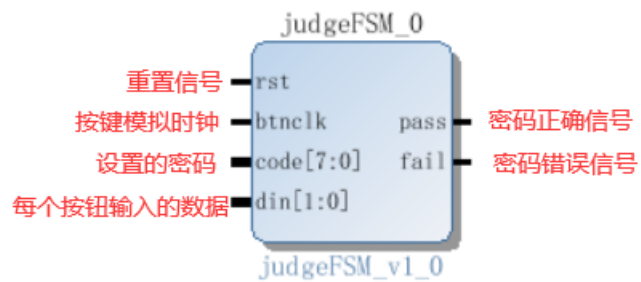
`btn[0]:2'00`;

btn[1]:2'01;

btn[2]:2'10;

btn[3]:2'11;

(3) 状态机检测模块



采用 Mealy 型状态机进行检测。

状态设置：

S0：初始状态，未接收密码

S1：前两位密码正确

S2：三、四位密码正确

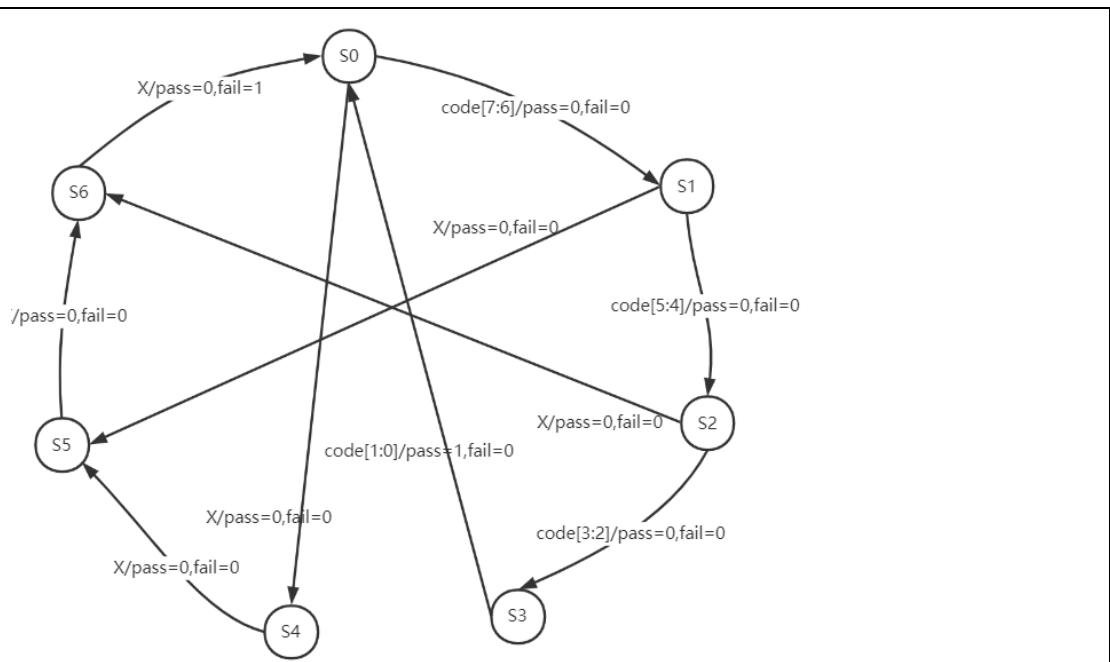
S3：五、六位密码正确

S4：前两位密码错误

S5：三、四位密码错误

S6：五、六位密码错误

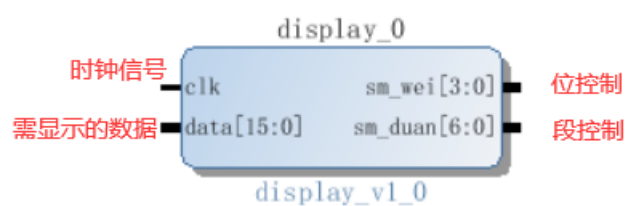
状态图：X 表示其他无效输入



状态表：

现态	次态					输出	
	code[7:6]	code[5:4]	code[3:2]	code[1:0]	X	!code[1:0]	code[1:0]
S0	S1	S4	S4	S4	S4	pass=0, fail=0	
S1	S5	S2	S5	S5	S5	pass=0, fail=0	
S2	S6	S6	S3	S6	S6	pass=0, fail=0	
S3	S0	S0	S0	S0	S0	pass=0, fail=1	pass=1, fail=0
S4	S5	S5	S5	S5	S5	pass=0, fail=0	
S5	S6	S6	S6	S6	S6	pass=0, fail=0	
S6	S0	S0	S0	S0	S0	pass=0, fail=1	

(4) 数码管显示模块



三、实验过程或算法

1、通过按键防抖使信号分别适用仿真和上板

```
// -----按键防抖-----
wire [3:0]btn;
if(flag) begin // 上板
    debkey debkey01(.clk(clk),.rst(rst),.button(button),.btn(btn));
end
else begin // 仿真
    assign btn = button;
end
```

dekey 模块内部：

设置自定义时钟，通过比较两次时钟上升沿保留的按键值，得到防抖后的按键模拟时钟。

```
module debkey(clk,rst,button,btn);
    input clk,rst;
    input [3:0]button;
    output [3:0]btn;

    wire myclk;
    integer f=32'd50_0000; // halfT = 5ms = 5e-3
    fdivider fdivider01(.clk(clk),.f(f),.myclk(myclk));

    // reg [4:0]last; // 也能成功防抖
    // always@(posedge myclk) last = button;
    // assign btn = (last==button)?last:5'b0;

    reg [4:0]key_rr,key_r;
    always@(posedge myclk,posedge rst) begin
        if(rst) begin
            key_r <= 5'b0;
            key_rr <= 5'b0;
        end
        else begin
            key_r <= button;
            key_rr <= key_r;
        end
    end
    assign btn = key_r & key_rr; // 总共延迟10+10 = 20ms
endmodule
```

2、通过 set 信号异步设置初始密码

```
//-----初始密码-----
reg [DW-1:0]code;
always(*) begin
    if(rst) code = {DW{1'b1}};
    else if(set) code = password;
end
```

3、在顶层模块中调用密码输入模块

```
//-----输入密码-----
wire [1:0]din;
wire [15:0]data;
wire btnclk; // 内置按键模拟时钟
inputData #(DW) inputData01(.rst(rst),.btn(btn),.btnclk(btnclk),.din(din),.data(data));
```

inputData 模块内部：

①通过按键输入数据

```
always@(posedge btn[0],posedge btn[1],posedge btn[2],posedge btn[3],posedge rst) begin
    if(rst) begin
        din <= 2'b00;
    end
    else begin
        if(btn[0]) din <= 2'b00;
        else if(btn[1]) din <= 2'b01;
        else if(btn[2]) din <= 2'b10;
        else begin din <= 2'b11; end
    end
end
//产生按键模拟时钟信号
assign btnclk = btn[0] | btn[1] | btn[2] | btn[3];
```

②通过拼接、移位等操作获得数码管需显示的数据

```
reg [3:0]cnt = 3'b0;
reg [15:0]temp;
// 这里需要捕获下降沿哦
always@(negedge btnclk,posedge rst) begin
    if(rst) begin
        cnt = 3'b000;
        temp = {16{1'b0}};
    end
    else begin
        if(cnt == 3'd4) begin
            cnt = 3'b000;
            temp = {16{1'b0}};
        end
        cnt = cnt + 1'b1;
        temp = (temp << 4) + {2'b00,din[1:0]}; // 这里不应该用非阻塞赋值
    end
end
assign data = temp;
```

4、在顶层模块中调用数码管显示模块

```
//-----结果显示-----
display display01(.clk(clk),.data(data),.sm_wei(res_wei),.sm_duan(res_duan));
```

5、在顶层模块中调用密码锁检测状态机

```
//-----密码判断-----
judgeFSM #(.DW(DW)) judgeFSM01(.rst(rst),.btnclk(btnclk),.code(code),
    .din(din),.pass(pass),.fail(fail));
```

judgeFSM 模块内部：

①次现态转换。根据状态转移图可得：

```

always@(negedge btnc1k,posedge rst) begin
    if(rst) begin
        curr = s0;
        next = s0;
    end
    else begin
        case(curr)
            s0: next = (din[1:0]==code[7:6])? s1:s4;
            s1: next = (din[1:0]==code[5:4])? s2:s5;
            s2: next = (din[1:0]==code[3:2])? s3:s6;
            s3: next = s0;
            s4: next = s5;
            s5: next = s6;
            s6: next = s0;
            default: next = s0;
        endcase
        curr <= next;
    end
end

```

②输出数据

```

always@(negedge btnc1k,posedge rst) begin
    if(rst) begin
        pass = 1'b0;
        fail = 1'b0;
    end
    else begin
        if(curr==s3 && din[1:0]==code[1:0]) begin //每一位都正确
            if (din[1:0]==code[1:0]) begin pass = 1'b1;fail = 1'b0; end
            else begin pass = 1'b0;fail = 1'b1; end
        end
        else if(curr==s6) begin pass = 1'b0;fail = 1'b1; end
        //检测最后一个按钮输入数据后给出fail信号
        else begin pass = 1'b0;fail = 1'b0; end
    end
end

```

五、实验过程中遇到的问题及解决情况

问题一：初始输入密码时，重置后即使输入密码正确也总是显示 fail

解决方案：修改状态机状态设计，使其状态转移情况符合预期

问题二：数码管无法如预期显示已经输入的密码，始终处于初始设置的

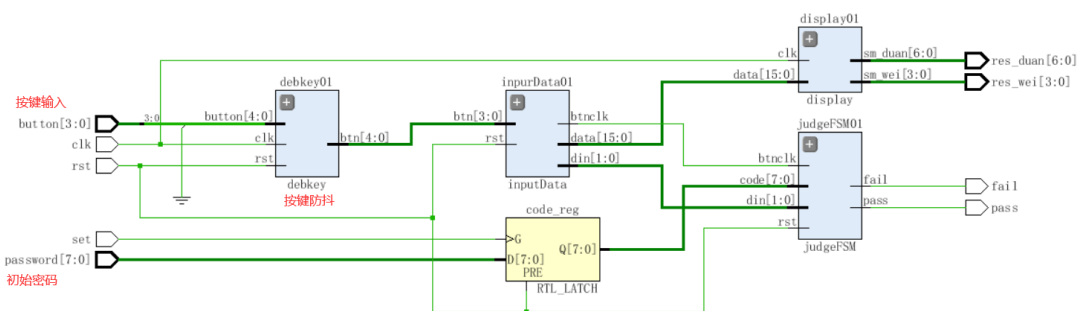
0000 的状态。

解决方案：经调试发现，应在 btnclk 的下降沿获得此时的 data 值，即密码管需显示的，已经输入后的密码，而不是在上升沿。修改后显示正常。

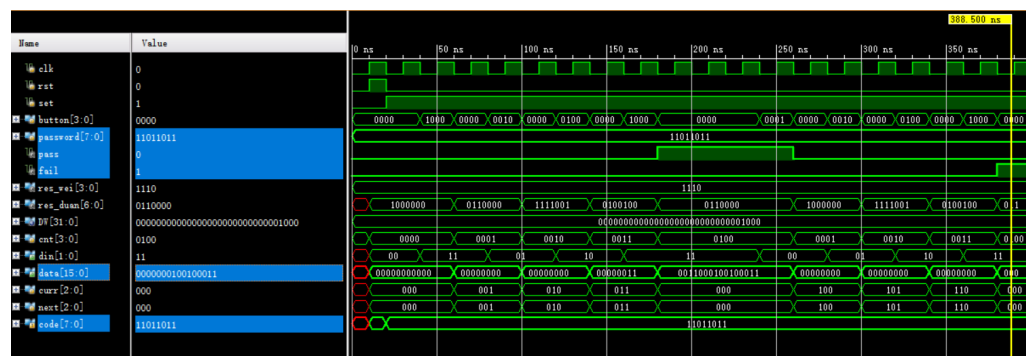
六、实验结果及分析和（或）源程序调试过程

1、RTL 电路图如图：

显示了分模块处理数据的思想



2、仿真时序图：符合预期



3、上板效果见附件视频

七、小组分工情况说明

李燕琴：顶层模块，状态机，数据输入模块编写

杨思怡：状态机，按键防抖模块编写