

# 数字逻辑与计算机组成实验

---

- 实验名称: lab7 状态机及键盘输入
- 院系: 计算机科学与技术系
- 学生姓名: 徐简
- 学号: 202220013
- 班级: 数字逻辑与计算机组成实验1班
- 邮箱: [161200063@smail.nju.edu.cn](mailto:161200063@smail.nju.edu.cn)
- 实验时间: 2022 年 4 月 20 日

# 数字逻辑与计算机组成实验Lab7

202220013 徐简 [161200063@smail.nju.edu.cn](mailto:161200063@smail.nju.edu.cn)

## 一, 实验目的

- 复习状态机的相关知识
- 利用状态机设计一个键盘输入控制器

## 二, 实验原理

- 有限状态机 FSM (Finite State Machine) 简称状态机, 是一个在有限个状态间进行转换和动作的计算模型。有限状态机含有一个起始状态、一个输入列表 (列表中包含所有可能的输入信号序列)、一个状态转移函数和一个输出端, 状态机在工作时由状态转移函数根据当前状态和输入信号确定下一个状态和输出。状态机一般从起始状态开始, 根据输入信号由状态转移函数决定状态机的下一个状态。
- PS/2是个人计算机串行 I/O 接口的一种标准, PS/2 接口可以连接 PS/2 键盘和 PS/2 鼠标。所谓串行接口是指信息是在单根信号线上按序一位一位发送的。
- 键盘通过 PS2\_DAT 引脚发送的信息称为扫描码, 每个扫描码可以由单个数据帧或连续多个数据帧构成。当按键被按下时送出的扫描码被称为“通码 (Make Code)”, 当按键被释放时送出的扫描码称为“断码 (Break Code)”。某按键的断码是 F0h 加此按键的通码, 如释放“W”键时输出的断码为 F0h 1Dh, 分两帧传输。

## 三, 实验环境/器材

- Quartus
- DE10-Standard开发平台
- FPGA开发板

## 四, 程序代码/流程图

手册中给出了ps2\_keyboard模块的代码, 该模块输出扫描码data以及ready、overflow等数据和信号。我们需要完成的核心部分是, 根据data得到按下的对应的按键。该部分封装在my\_keyboard模块中。

代码如下, 只展示与核心功能有关的变量和功能实现

```
module my_keyboard(  
    );  
    ps2_keyboard inst(  
//initiate ps2_keyboard module  
    );  
  
    always @ (posedge clk) begin  
        if (clr_n == 0) begin  
            //clear all signals  
            nextdata_n <= 1;  
            pressed <= 1;  
            has_counted <= 0;  
            eff_data <= 8'b0;  
            keys_times <= 0;
```

```

        end else begin
            if (ready) begin
                if (data == 8'hF0) begin // 断码F0
                    pressed <= 0; // 松开
                    if (has_counted == 0) begin // 避免按住重复数
                        keys_times <= keys_times + 1; // 次数加1
                        has_counted <= 1;
                    end
                end else begin
                    has_counted <= 0;
                    if (pressed == 0) begin
                        pressed <= 1;
                    end else begin // 有效的数据
                        eff_data <= data;
                    end
                end
            end
            nextdata_n <= 0; // 设置指针前移信号
        end else nextdata_n <= 1;
    end
end
end

endmodule

```

这样就完成了最核心的功能，剩下的功能有：

- 显示建码、ASCII码以及按键总次数：实例化三个七段数码管即可，不再赘述
- 按键松开时，七段数码管第四位全灭

```

if (alo)
    begin
        ascii_h <= 7'bz;
        ascii_l <= 7'bz;
        scancode_h <= 7'bz;
        scancode_l <= 7'bz;
    end
end

```

自己选择一个合适的时间长度，通过计数到该长度段且没有新数据的话，说明什么按键都没有，这时候让数码管熄灭，下面的模块，输入是时钟，ps2\_keyboard输出的raw数据，以及输出熄灭灯信号。

```

module all_lights_out(clk, ps_data, alo);
    input clk, ps_data;
    output reg alo;
    integer cnt,gt;
    //cnt计数器，gt自己设定的判断没有按键的时间间隔
    initial begin
        cnt = 0;
        alo = 0;
        gt=1000000;
    end

    always @ (posedge clk) begin
        if (ps_data == 0) begin
            cnt <= 0;
            alo <= 0;
        end else begin

```

```

        if (alo == 1 || cnt == gt) begin
            alo <= 1;
        end else begin
            cnt <= cnt + 1;
            alo <= 0;
        end
    end
end
end

endmodule

```

- 实现键码和ASCII码的转换，将该功能封装到一下模块，其中用txt文件初始化转换表，剩下的工作就是每次查表输出对应值即可

```

module key_to_ascii( addr, dout_vec);
    input [7:0] addr;
    output reg [7:0] dout_vec;
    reg [7:0] dic[255:0];
    initial begin
        dout_vec = 8'b0;
        $readmemh("D:/My_design/lab7/scancode.txt", dic, 0, 255);
    end
    always @ (*) begin
        dout_vec = dic[addr];
    end
end
endmodule

```

以上完成了必做部分的设计，选做部分目前还没有全部实现，只预留了大小写锁定指示灯，具体逻辑未完成，应该会在验收前补上。

## 五, 实验步骤/过程

- 使用DE10软件建立工程，这样可以自动进行引脚分配
- 编写代码，并在项目中实例化
- 进行测试
- 上板验证

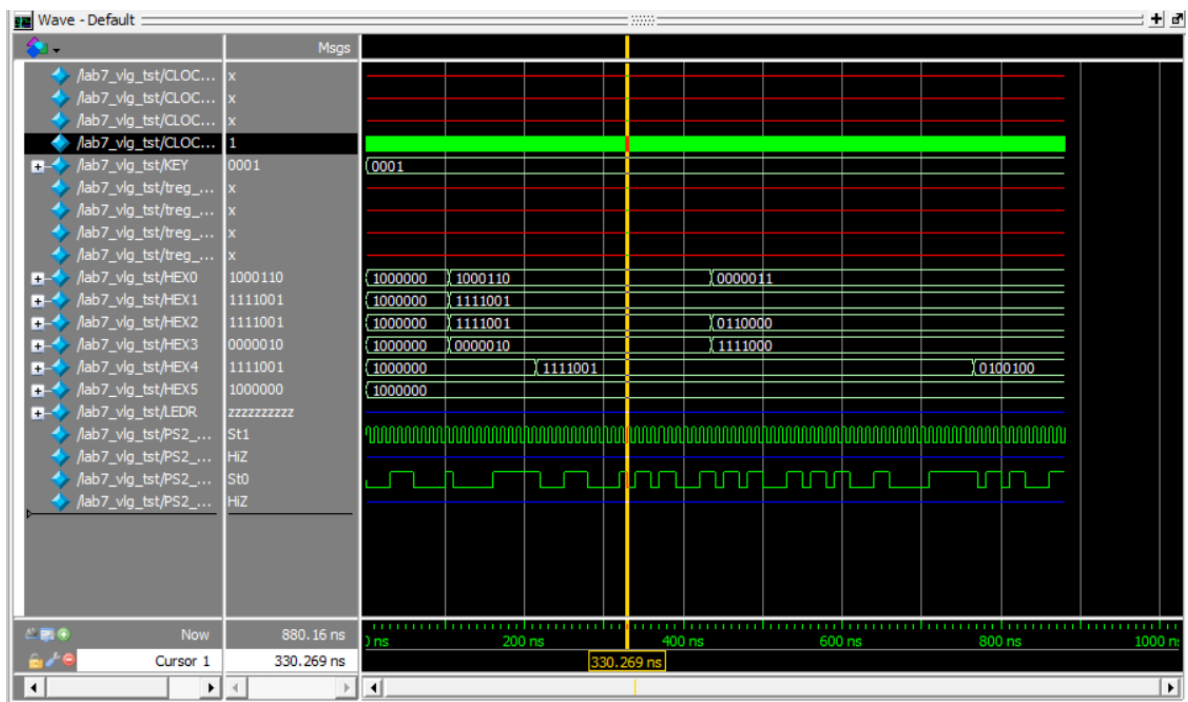
## 六, 测试方法

- 仿真：按照手册推荐方法
- 上板测试：写入fpga开发板，根据功能测试输入输出的情况

## 七, 实验结果

仿真

这里LEDR显示大小写锁定的功能还没有实现



上板（尚未完成，还没拿到键盘）

## 八, 遇到的问题和解决办法

- 这次的仿真比较复杂（Quartus操作上），需要将两个文件都添加进testbench

## 九, 实验得到的启示

- 要早点开始实验orz，不然会做不完（错误的估计了复杂程度）

## 十, 意见和建议

- 希望提供的测试代码可以覆盖选做内容