**中国科学技术大学计算机学院**

**《数字电路实验》报告**



实验题目：信号处理及有限状态机

学生姓名：张郑飞扬

学生学号：PB21071416

完成日期：2022.12.1

计算机实验教学中心制

2020年09月

【实验题目】

信号处理及有限状态机

【实验目的】

进一步熟悉 FPGA 开发的整体流程

掌握几种常见的信号处理技巧

掌握有限状态机的设计方法

能够使用有限状态机设计功能电路

【实验环境】

VLAB： vlab.ustc.eud.cn

FPGAOL:fpgaol.ustc.edu.cn

Logisim

Vivado

【实验过程】

**Step1.信号整形及去毛刺**

阅读实验手册，了解了信号整形的基本原理是通过一个计数器对高电平持续时间进行计时，当按键输入信号为 0 时，计数器清零，当输入信号为高电平时，计数器进行累加计数，计数达到阈值后则停止计数。通过调整阈值可以改变电路精度。

**Step2.取信号边沿技巧**

我们知道，除了时钟和异步复位信号外，其它信号都不应放在边沿敏感列表内。那么怎么用其他信号边沿来触发状态跳转呢？实验手册给出了下面的方法：

*always@(posedge clk)*

*if(控制信号==1)*

*//状态跳转*

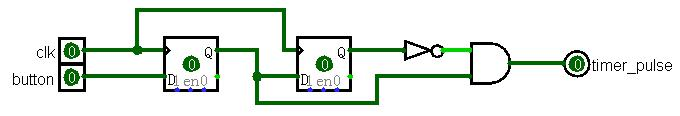
*else*

*//状态保持*

上述代码能够实现电路状态转换的功能，但是要求该信号为持续时间为一个时钟周期的脉冲信号， 否则电路状态会一直跳转。而人的反应时间相对于频率快的时钟来说是很慢的，因此要求该信号为持续时间为一个时钟周期的脉冲信号并不现实。

下面学习通过该按键信号生成一个时钟周期宽度的脉冲信号：

在Logism中画出电路图如下



其 Verilog 代码如下：

*module signal\_edge(*

*input clk,*

*input button,*

*output button\_edge);*

*reg button\_r1,button\_r2;*

*always@(posedge clk)button\_r1 <= button;*

*always@(posedge clk)*

*button\_r2 <= button\_r1;*

*assign button\_edge = button\_r1 & (~button\_r2);*

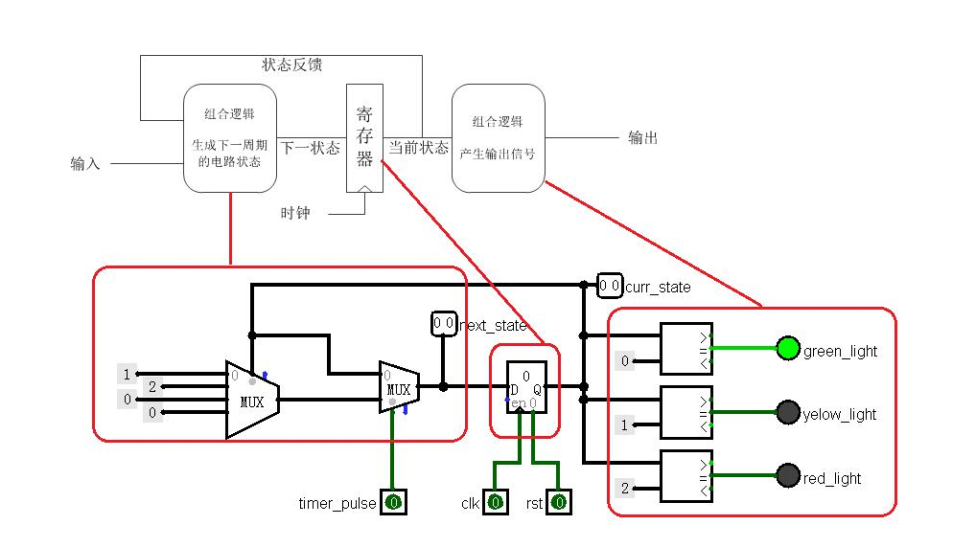
*endmodule*

该信号在 button信号的上升沿附近为高电平，其余时间均为低电平。解决了上面提出的问题。

**Step3、4.有限状态机介绍及实现**

对n位位宽的寄存器，状态不会超过2^n个，即其状态数量是有限的，因此这种电路结构称为有限状态机。有限状态机分为摩尔型和米利型。摩尔型的输出信号仅与当前状态有关，而米利型不仅与当前状态有关，还与输出信号有关。

关于FSM的知识已经在数字电路课程上学习过，此处不多作赘述。

实验手册上给出了一个控制交通信号灯的有限状态机。给出了Verilog代码，并在Logism中对其进行了实现。如下图：

FSM的Verilog代码有三种写法：一段式、两段式、三段式。三段式的可读性最好，实验手册也推荐使用三段式写法。

**Step5. 有限状态机的另一形式**

实验手册在该部分给出了一个电路，它在前面实验就出现过，虽然不是按三段式来写的，但确实是一个有限状态机。有组合部分也有时序部分。

【实验练习】

**题目1.**

Verilog代码如下：

module test(

input clk,rst,

output led);

reg [1:0] current\_state

reg [1:0] next\_state

//第一部分

always@(\*)

begin

next\_state = current\_state + 2’b01;

end

//第二部分

always@(posedge clk or posedge rst)

begin

if(rst)

current\_state <= 2’b00;

else

current\_state <= next\_state;

end

//第三部分

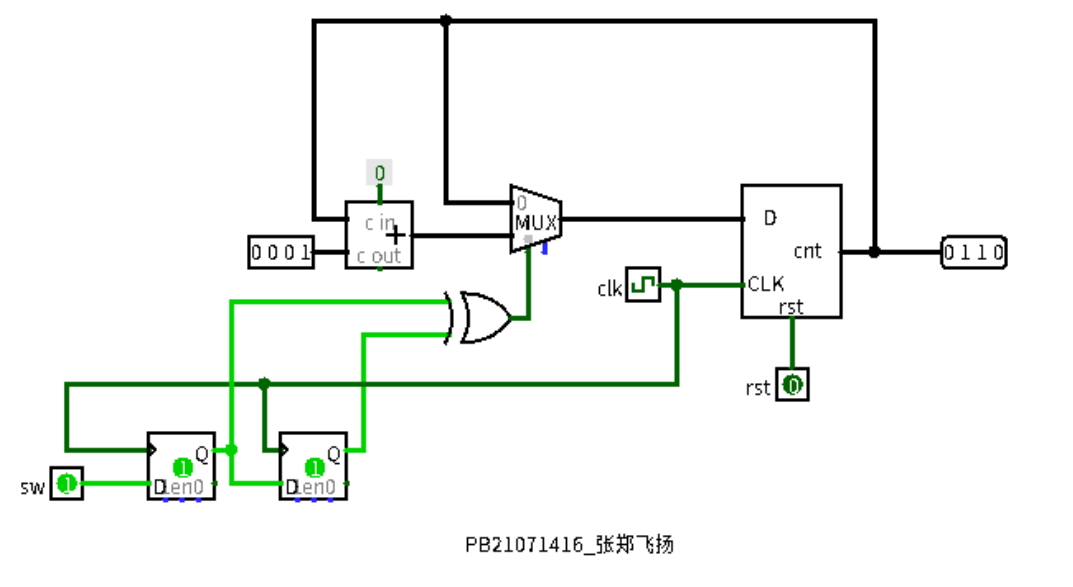
assign led = (current\_state == 2’b11)? 1’b1 : 1’b0;

endmodule

**题目2.**

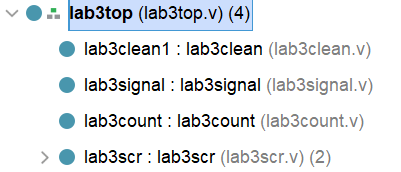
本题思路为，使用Step2中的取边沿技巧，在sw的上升沿和下降沿附近都输出一个一时钟周期的高电平信号。

为此，我们需要对step2中的电路进行一些改造，即使用一个异或门代替step2电路中的与门，从而实现在下降沿附近也能输出高电平信号的功能。

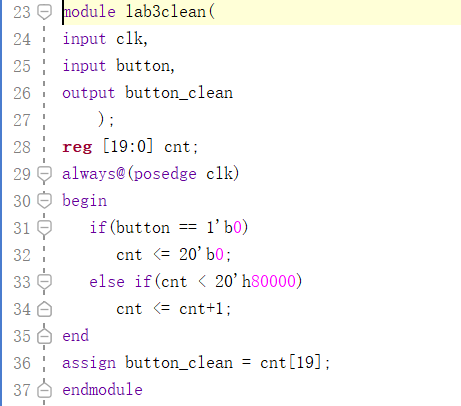
具体电路图如下图所示：

**题目3.**

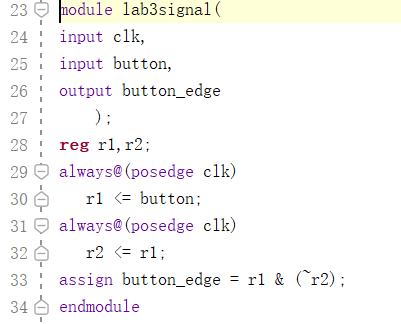
将该FSM分为如下几个模块：



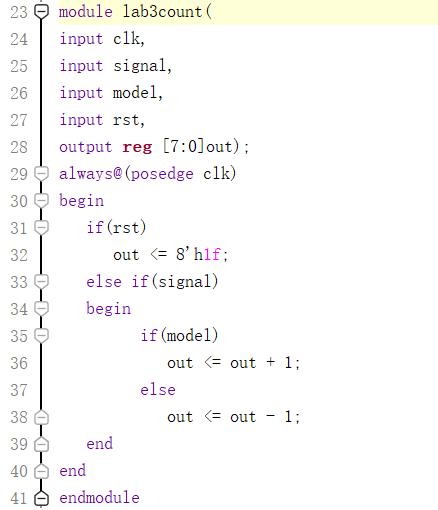
其中，clean模块是去毛刺模块，原理同step1，代码如下：



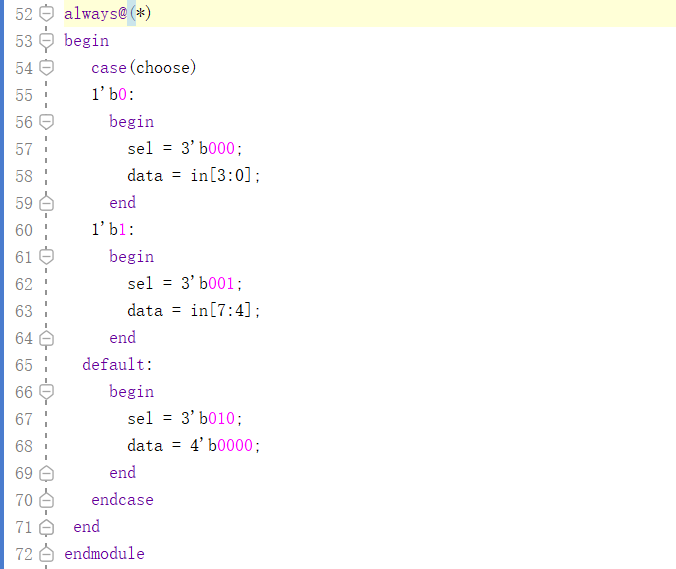
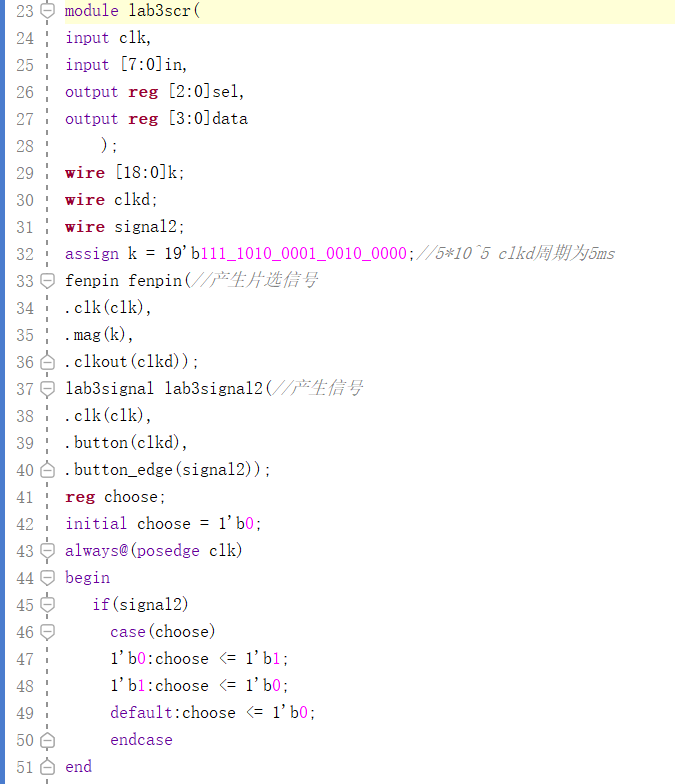
signal是产生边缘信号模块，原理同step2，代码如下：



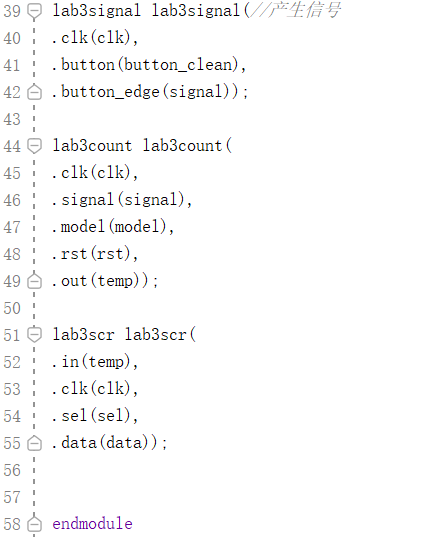
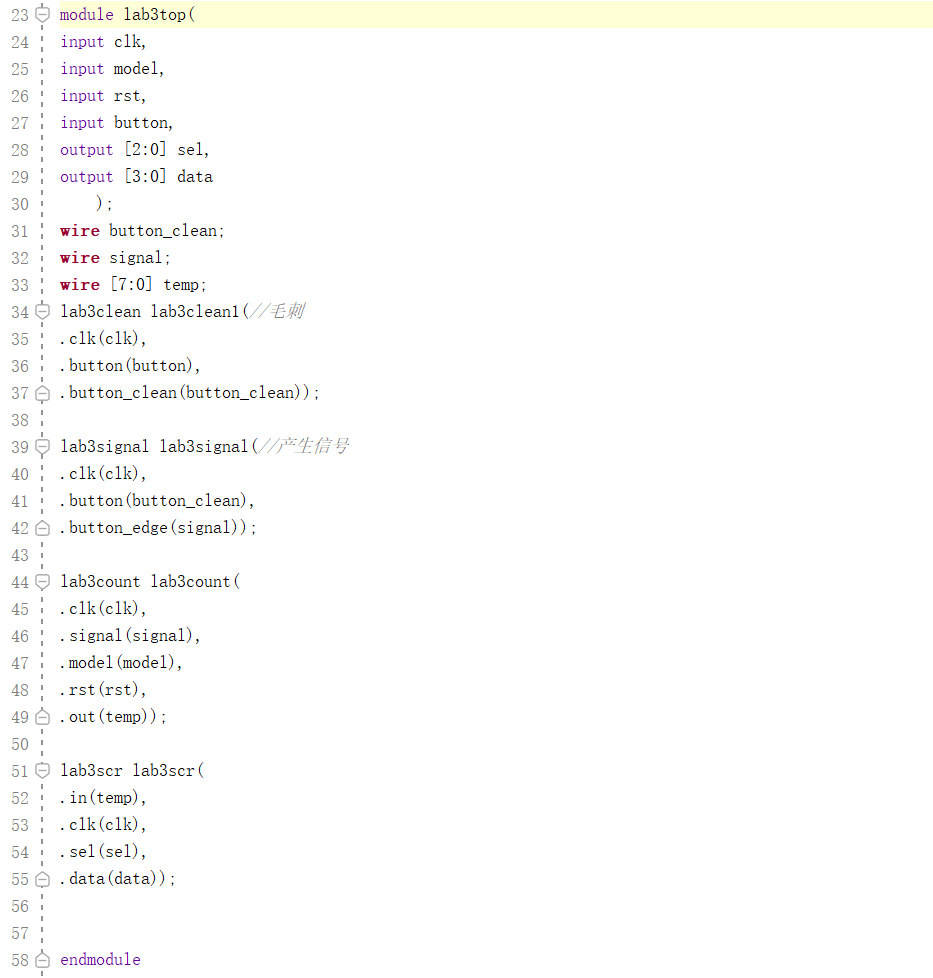
count模块是计数器模块，其实现比较简单，代码如下：



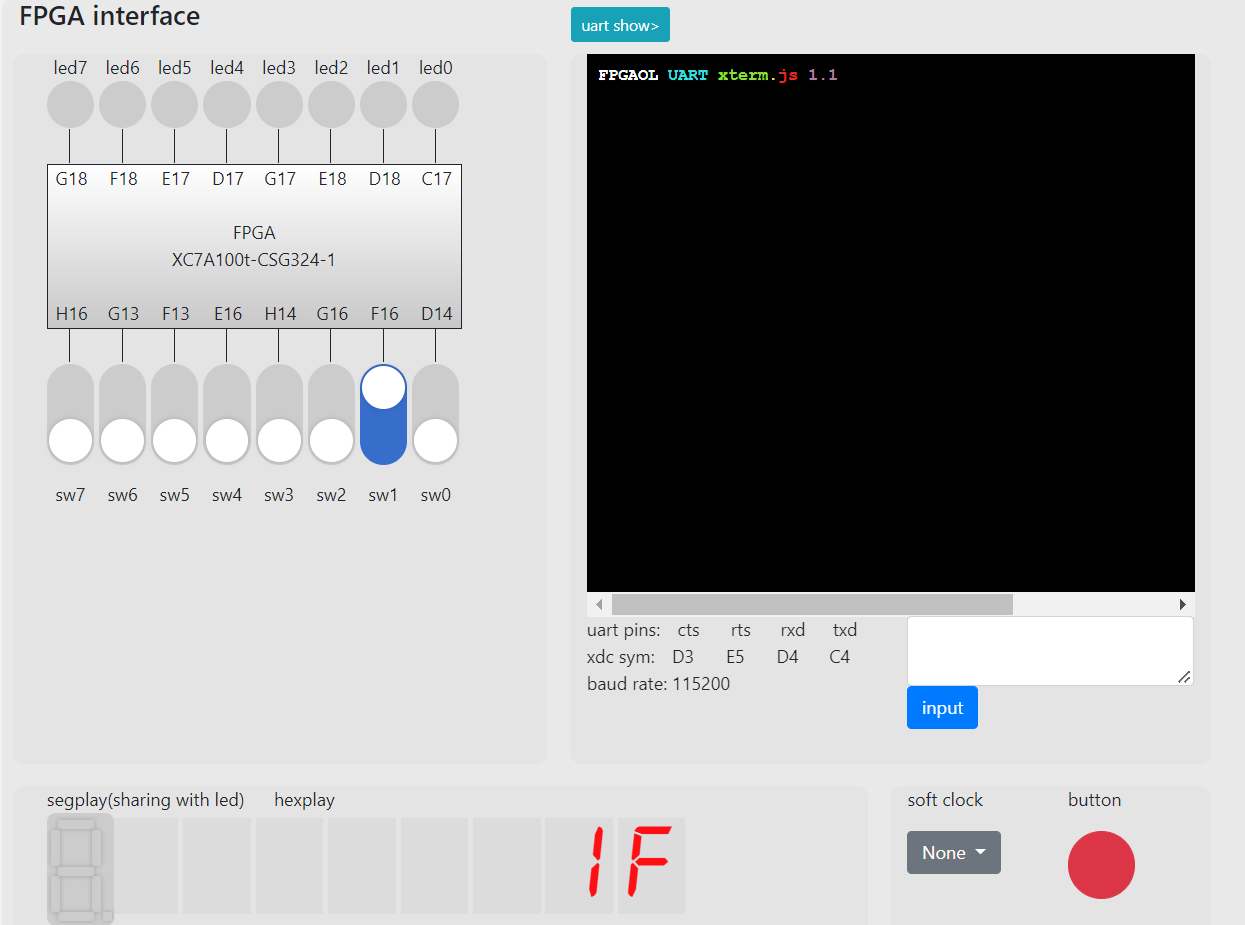
此外，为了实现在数码管上的呈现，需要和实验七中一样采用时分复用的方法，其中分频等操作的原理都和实验七中的原理一样。不再赘述。另外，信号的产生也需要使用上面signal模块里的取信号边沿功能。综上，我们可以设计出scr模块，其代码如下：



有了以上这些模块，我们就可以实现该计数器了，其顶层代码如下：

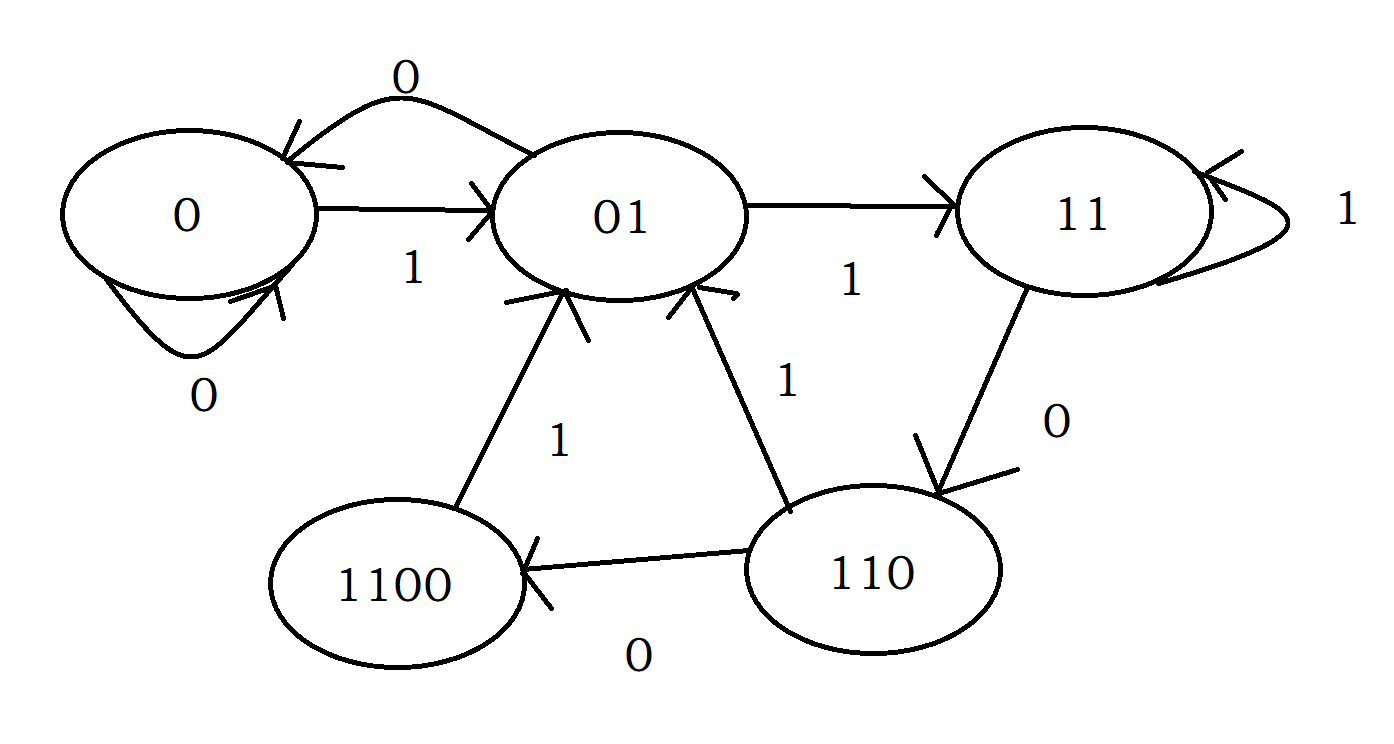


烧写到FPGA上，成功实现了所需功能：

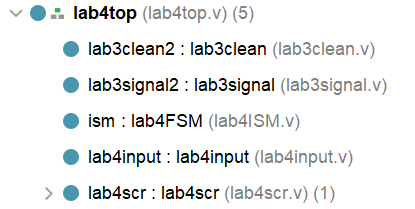


**题目 4.**

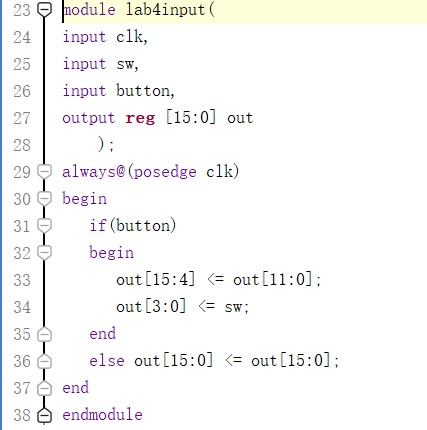
对于该FSM，其状态图如下：



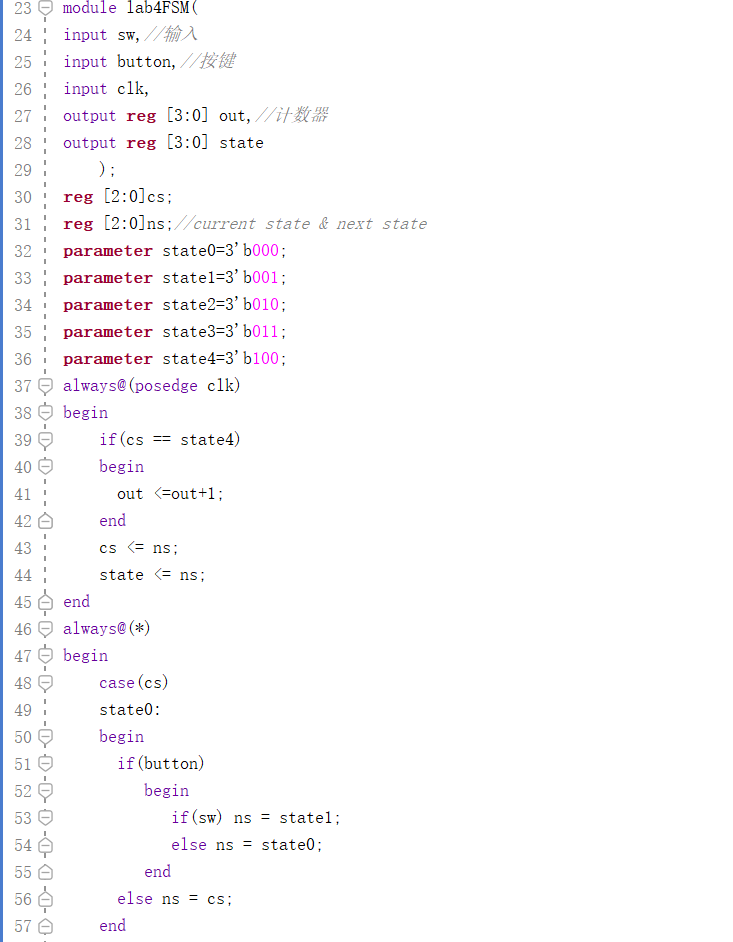
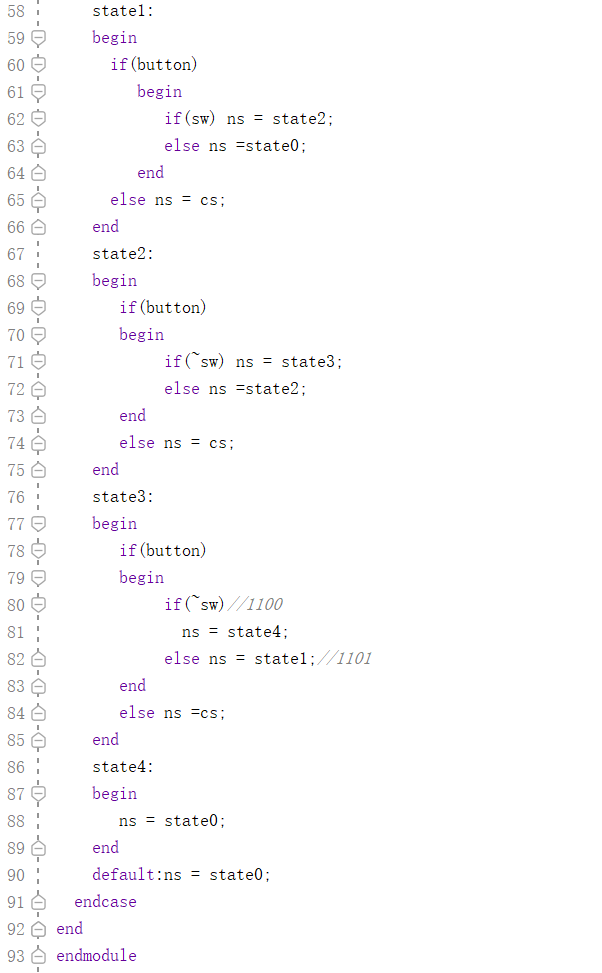
接下来，分模块设计该FSM，分析得知，类似题目3，我们需要去毛刺模块、取信号边沿模块、数码管的时分复用模块，此外，我们还需要设计一个输入模块，在该模块中，对数据进行移位处理，根据button是否被按下来决定数据是否移位，并产生新的4位数据。于是，模块的分配如下图所示：



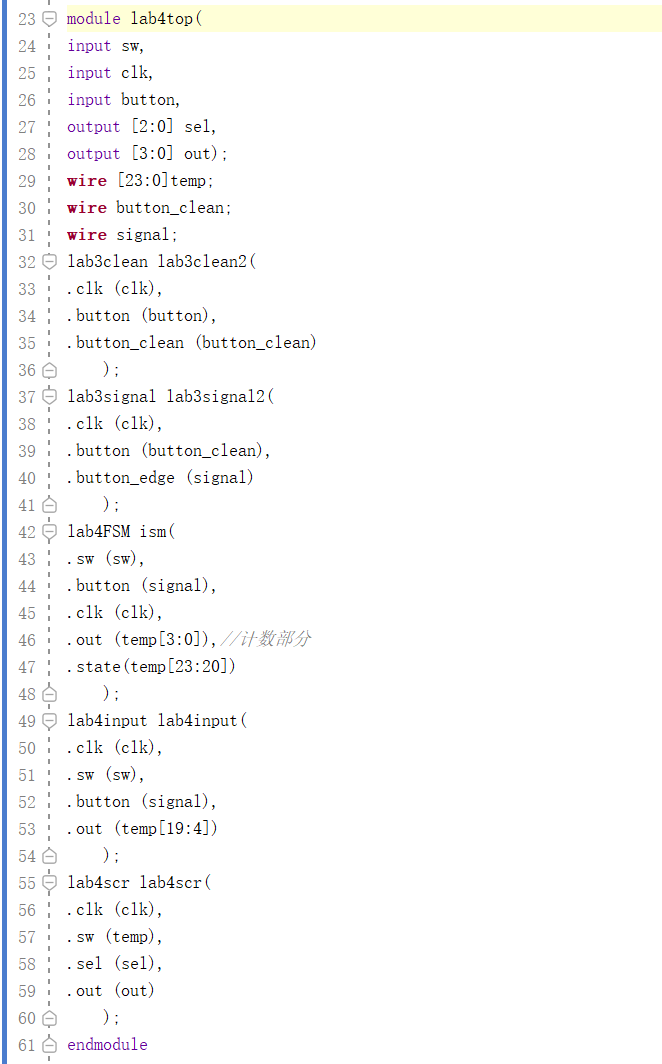
对去毛刺模块（clean）和取信号边沿（signal）模块不再赘述，和题目3大体相同。对数码管的时分复用模块（scr），我们规定低4位为计数器，中16位为前四次sw，高4位为当前状态，用六个数码管显示全部所需数据。以上三模块的代码都和题目三差不多，不再展示。接下来展示input模块，该模块也比较简单：



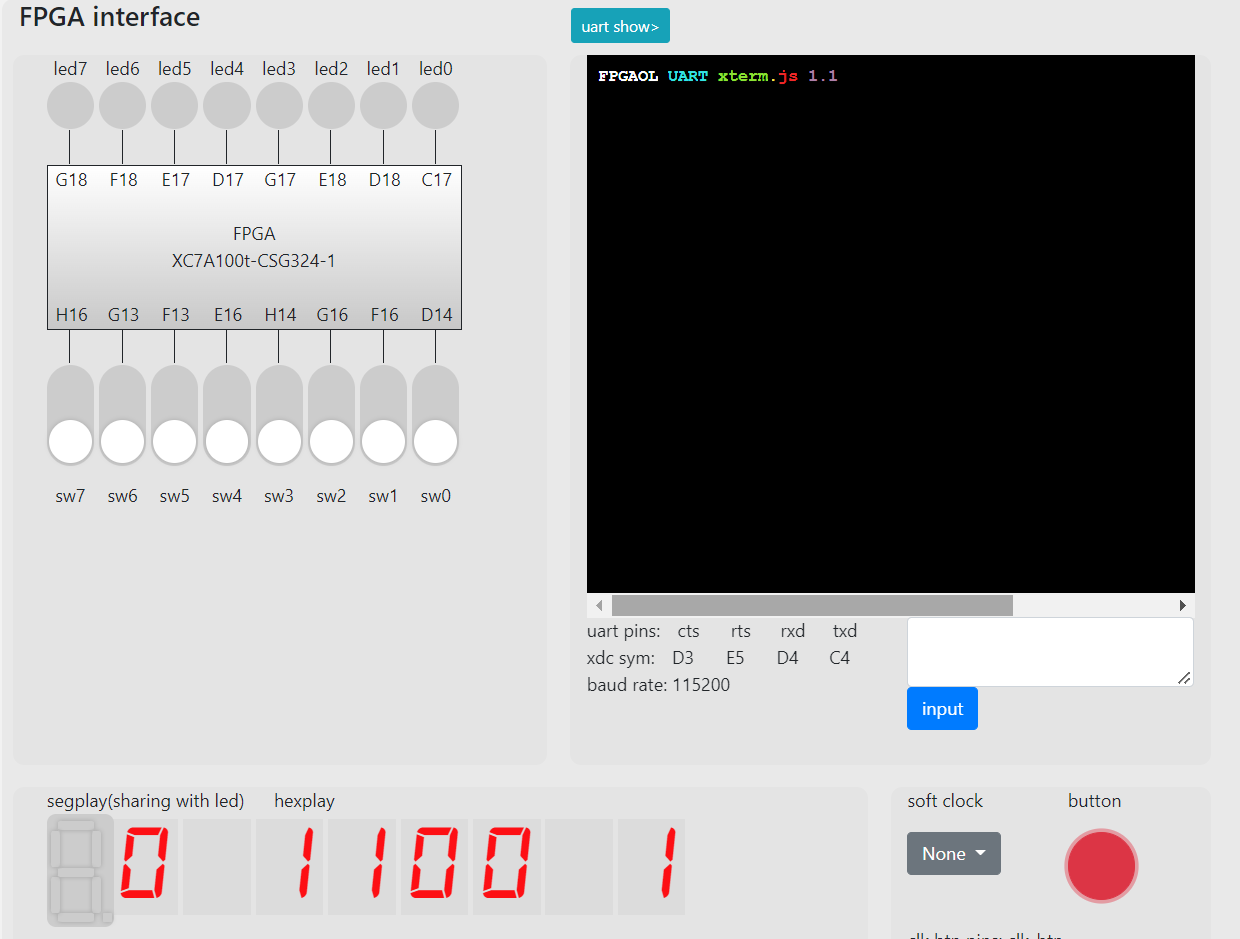
有了这些模块，我们就可以构建状态转换模块（FSM）：

最终，顶层代码如下图所示：



烧写到FPGA上，成功实现了所需功能：



【总结与思考】

本次实验主要系统学习了有限状态机的Verilog编码实现，并且学习了一些处理信号的技巧，比如去毛刺和取信号边沿，这些技巧对设计电路起到了很大的帮助。