

数字逻辑实验三 实验报告

**学 院： 计 算 机 工 程 学 院**

**班 级： 计算2114**

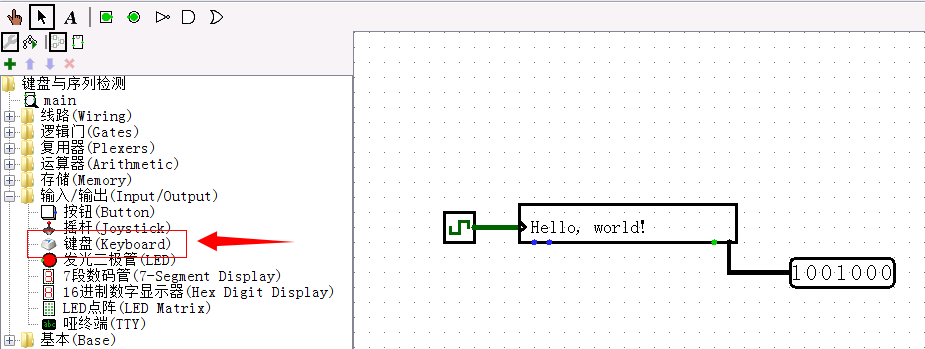
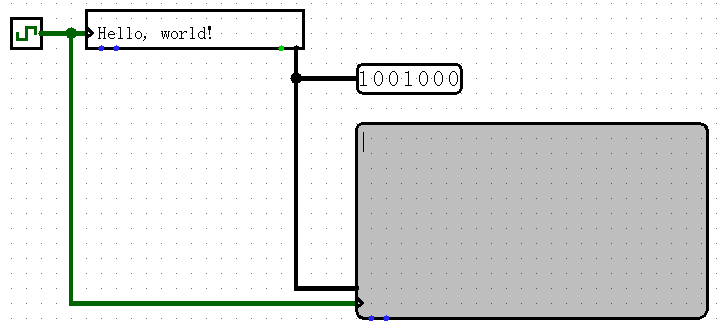
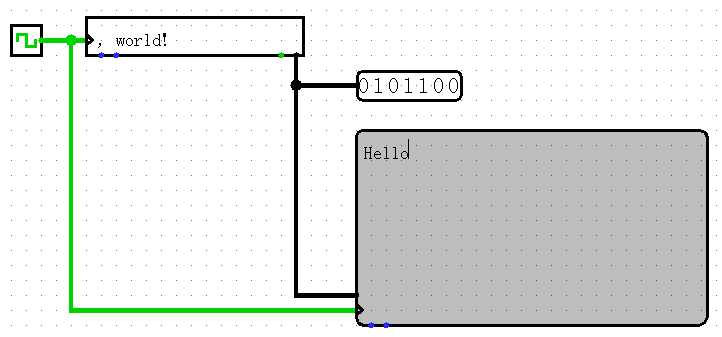
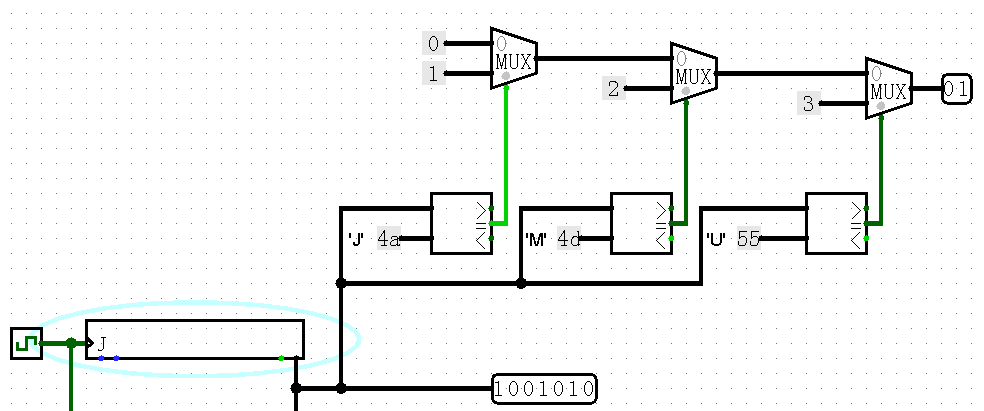
**姓 名: 庄佳强 章立早**

**学 号： 202121331104 20212133118**

2022 年 4 月 26日

# 实验二.c

# 一 题目要求

**从键盘的输出中检测特定字符串** Logisim提供一个名叫“键盘”的输入模块，允许你在缓冲区当中输入一个字符串，并且时钟的触发下，顺序地按字符的ASCII码输出，这个过程如下图所示：  
  
  
 为了进一步丰富你的工具箱，我们将使用Logisim提供的一个名为“哑终端”的输出模块（就在“键盘”的不远处）。如下面两幅图所示，哑终端能够在时钟的触发下，将接收到的ASCII码顺序地显示出来：  
  
  
 在这个任务当中，你需要使用有限状态机的方法，设计一个能够从键盘输出序列中检测“JMU”字符串的系统，并且当你成功检测到“JMU”字符串，在哑终端上打印出一个字符‘V’（ASCII码为‘0x56’）。  
 为了能够进一步简化有限状态机的设计，我们可以对输入的字符进行编码，而不是使用原始7比特ASCII值作为输入。比如，你可以参考如下设计，分别将‘J’编码为012、‘M’编码为102、‘U’编码为112、其它字符编码为002：  


# 二 设计思路

1使用有限状态机，设计不同字母下的不同编码，再转化编译出来。

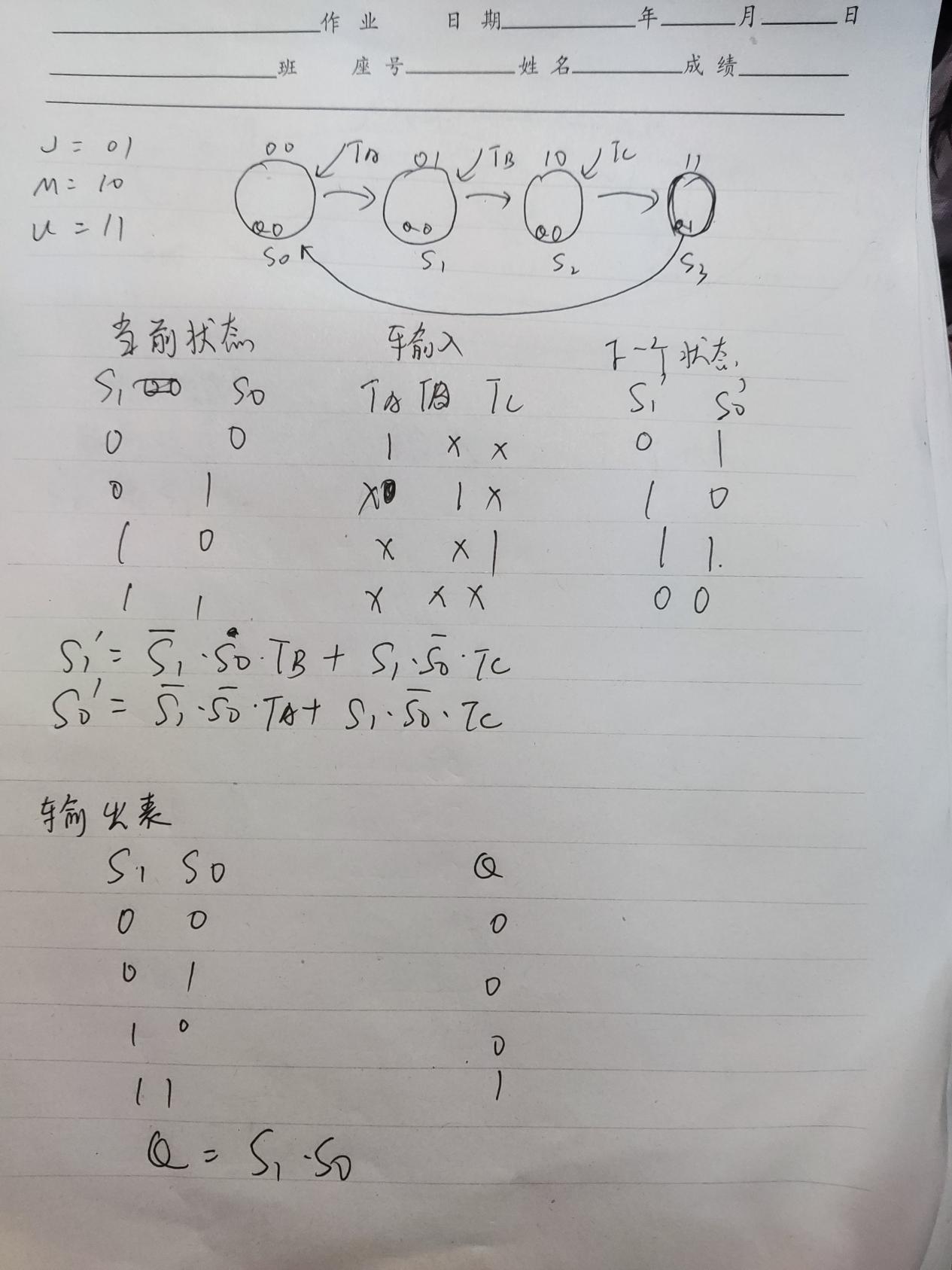
# 三 成果展现

**输入变量：**S1,S0TA,TB,TC

**状态个数：**4

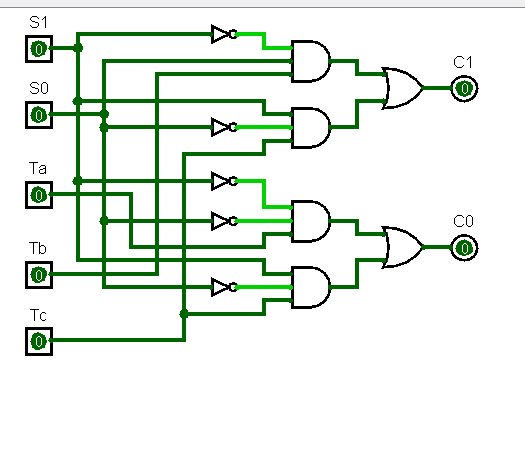
**状态图，状态转换表和输出表**

**逻辑表达式的推导**

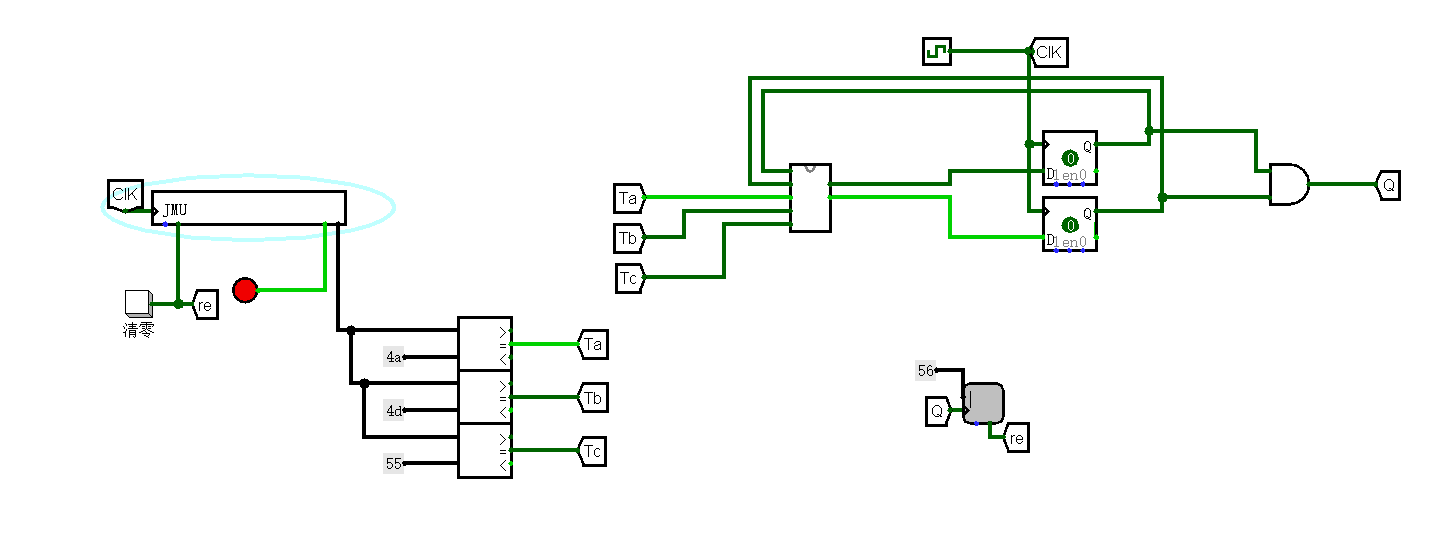


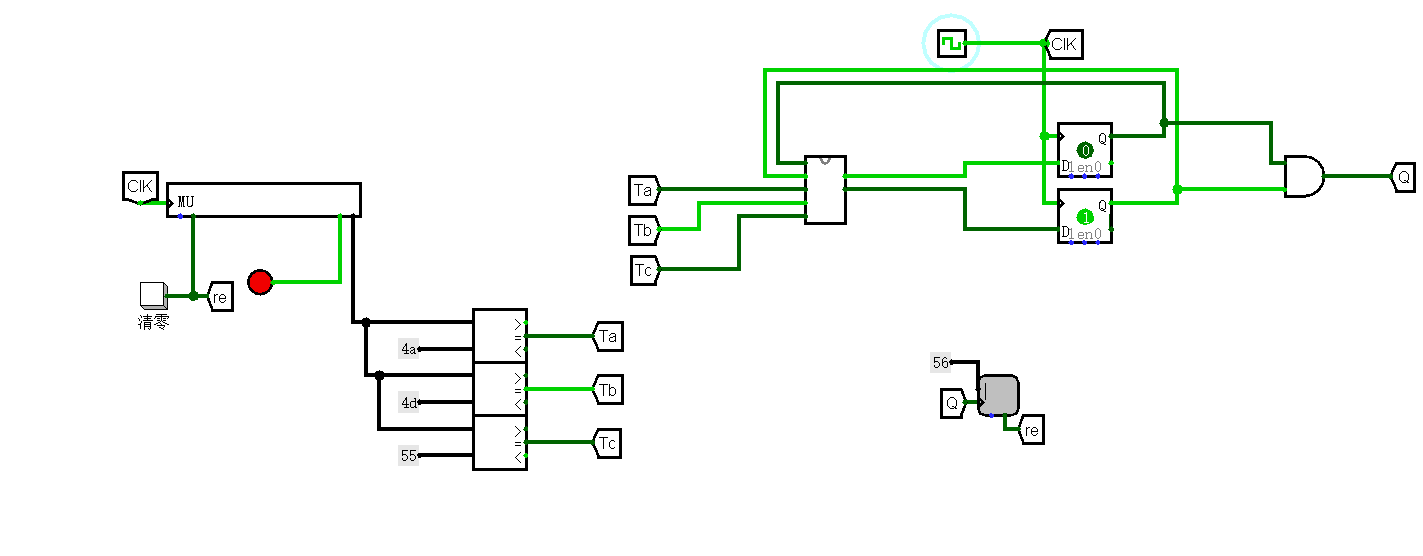
**电路图：**

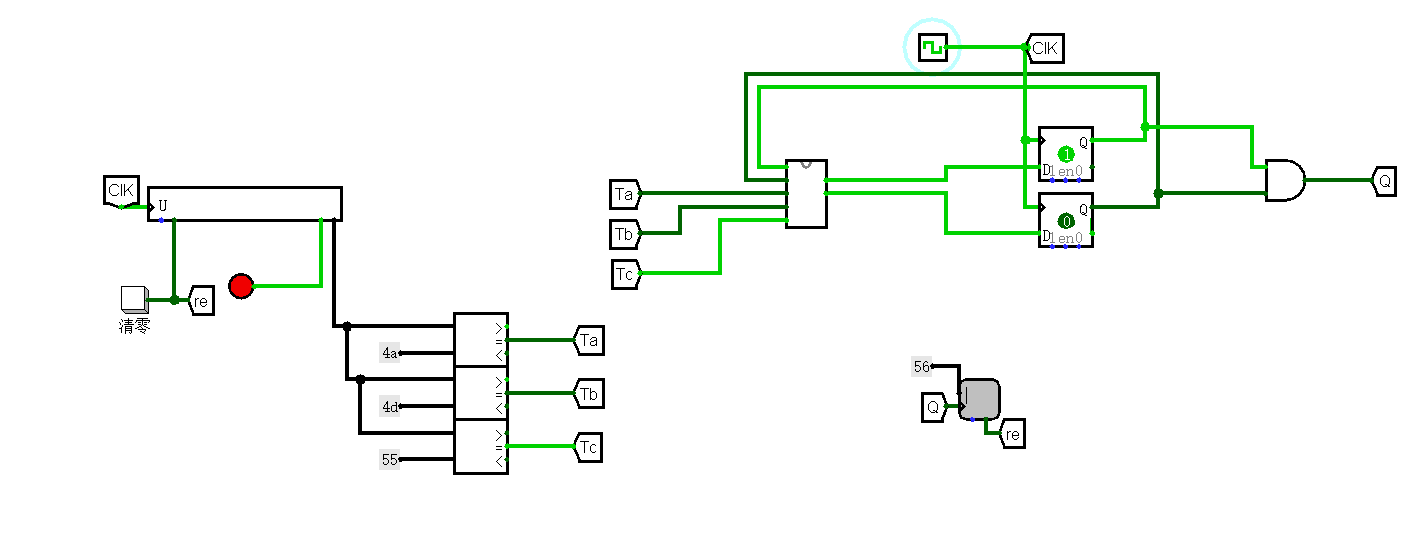
**有限状态机部分：**

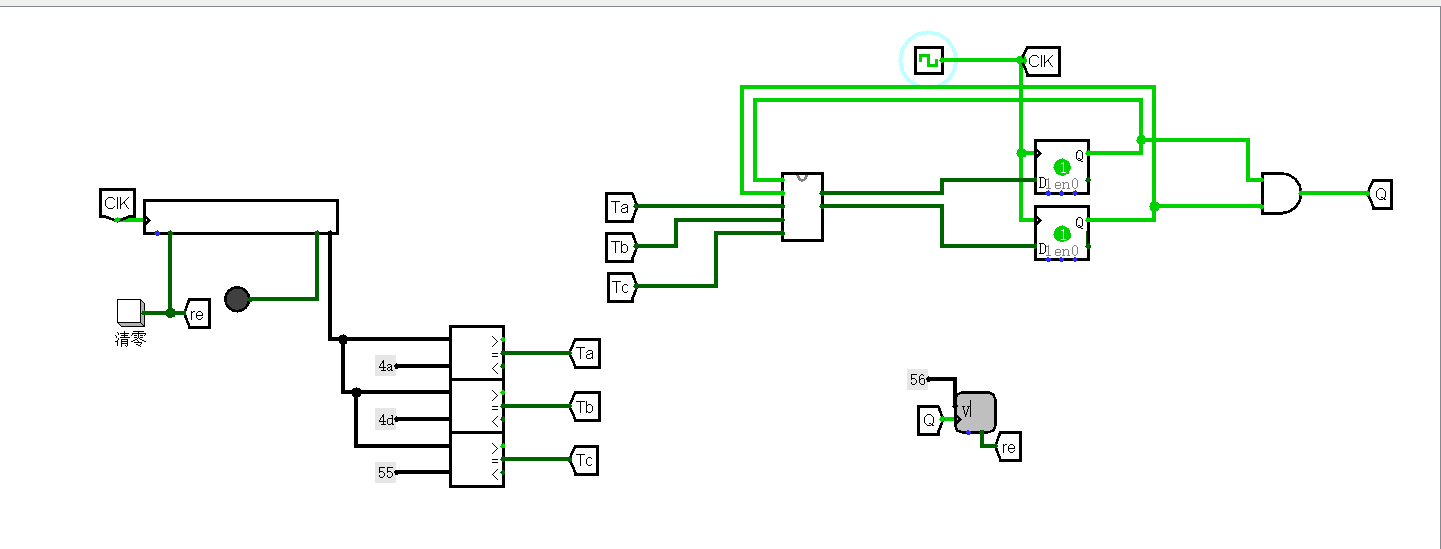
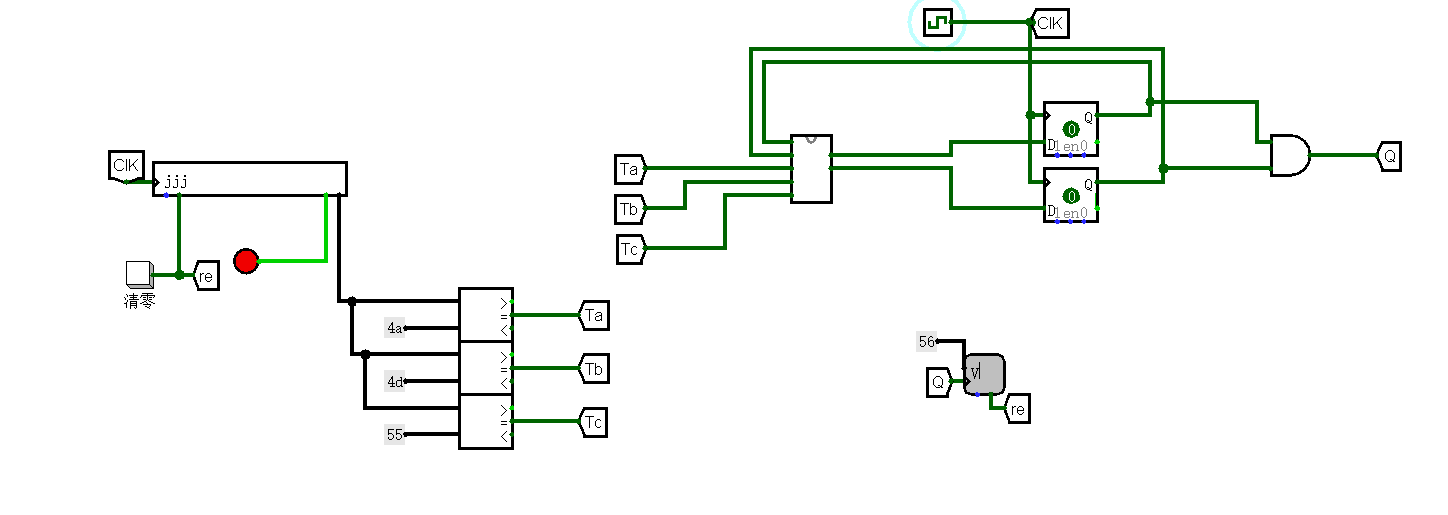


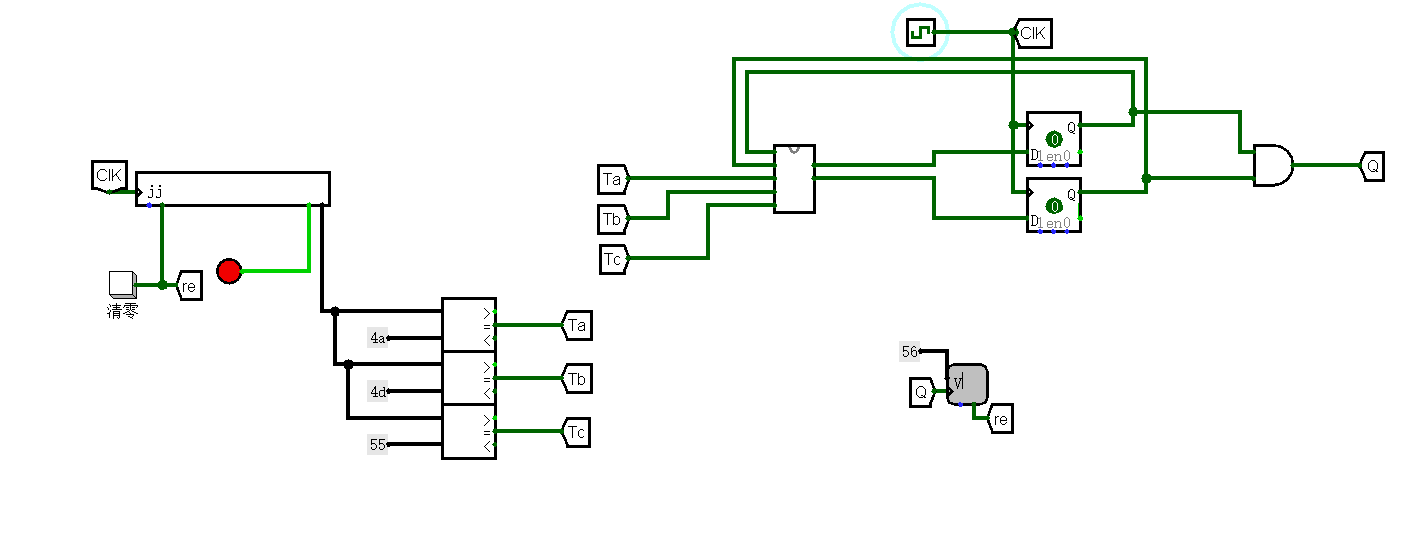
**主体部分：**



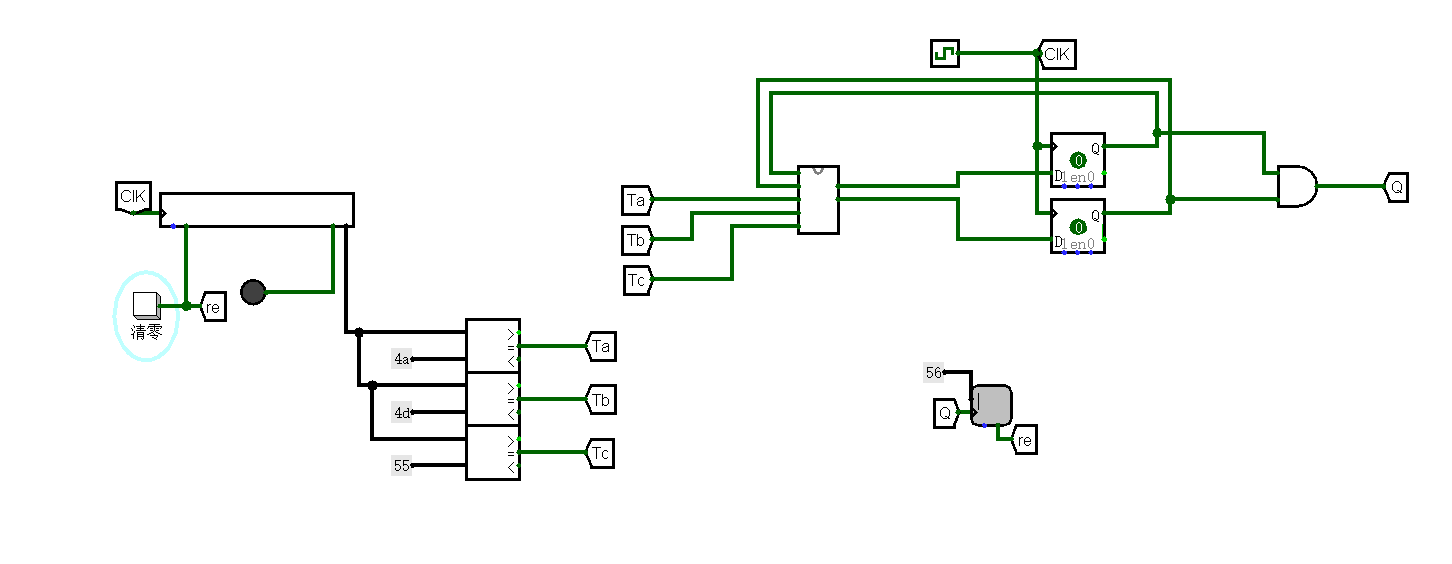






**按下清空后**



# 四 实验细节

需要使用D触发器来存储数据，来返回使用。

# 五 实验总结

巩固了有限状态机的使用，学会使用隧道，键盘等一系列封装插件。

# 实验一.D

# 一 题目要求

**生活中存在很多应用可以用有限状态机的思想去描述和实现。比如电梯当中的按钮：如果你很快速地按一下按钮，电梯将会把你希望去的楼层记录下来；如果你长时间地按同一个按钮，某些电梯又会取消你过去的输入记录。在这个任务中，你需要根据自己对生活的观察或体验，寻找一个合适的应用场景中的问题，并采用有限状态机的方法加以解决。当你完成设计之后，你还需要在Logisim或其它平台中完成相应的实现和测试。**

**1.应用场景：**

将参与为部门休息室设计一个苏打汽水自动售货机。苏打水项目由ⅠEEE 的学生分会部分资助的,因此它们的价格仅为25美分。机器接收5美分、10美分和25美分硬币。当投入足够的硬币时苏打汽水自动售货机就会分配汽水和找零钱。为这个苏打汽水自动售货机设计一个有限状态机控制器。有限伏态机的输入是 Nickel(5美分)、Dime(10美分)和 Quarter(25美分）,表示硬币已经投入机器。假设在一个周期投一个硬币。当有限状态机达到25美分分时,它将给出Dispense汽水和相应的 Return 输出,需要给出合适的找零。接着它准备开始接收硬币以便售下一瓶汽水。

# 二 设计思路

1. **输入变量:**S2,S1,S0,TA,TB,TC
2. **状态个数:**7

使用有限状态机设计化数字为状态，每一个钱化成一个状态。

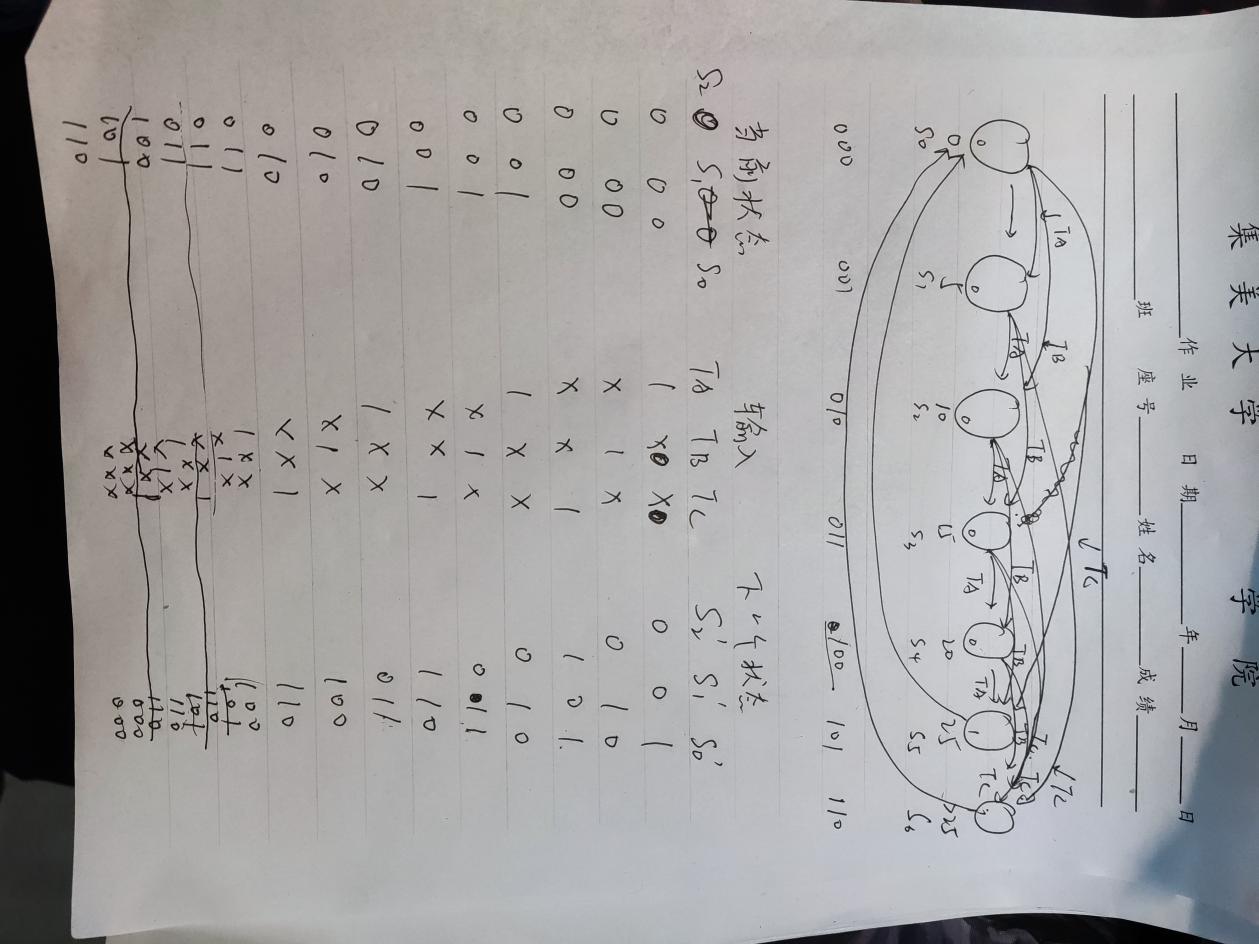
用加法器来记录当前金钱，当大于等于25时，用减法器减去，得出余额，用16位数码管输出出来。

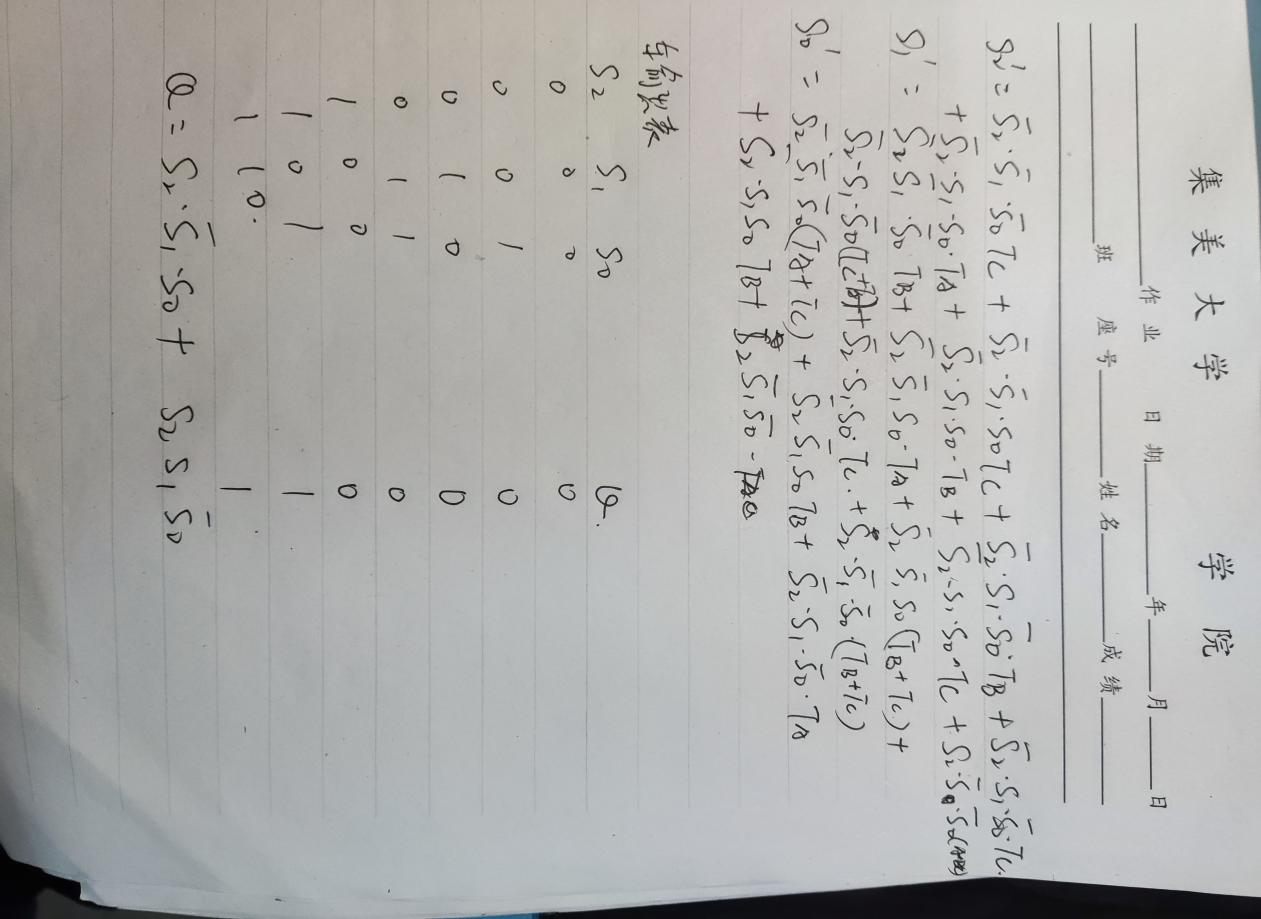
# 三 成果展现

状态图

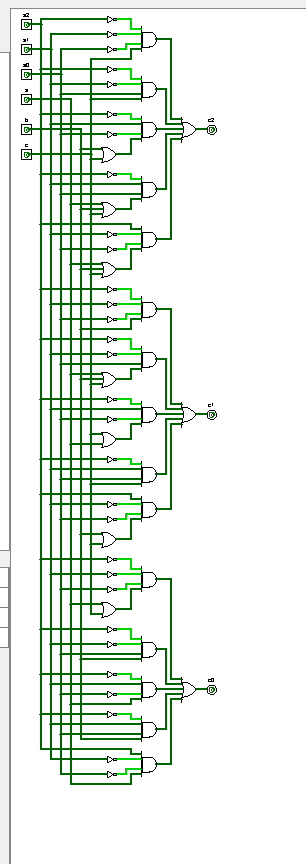
状态转换表和输出表

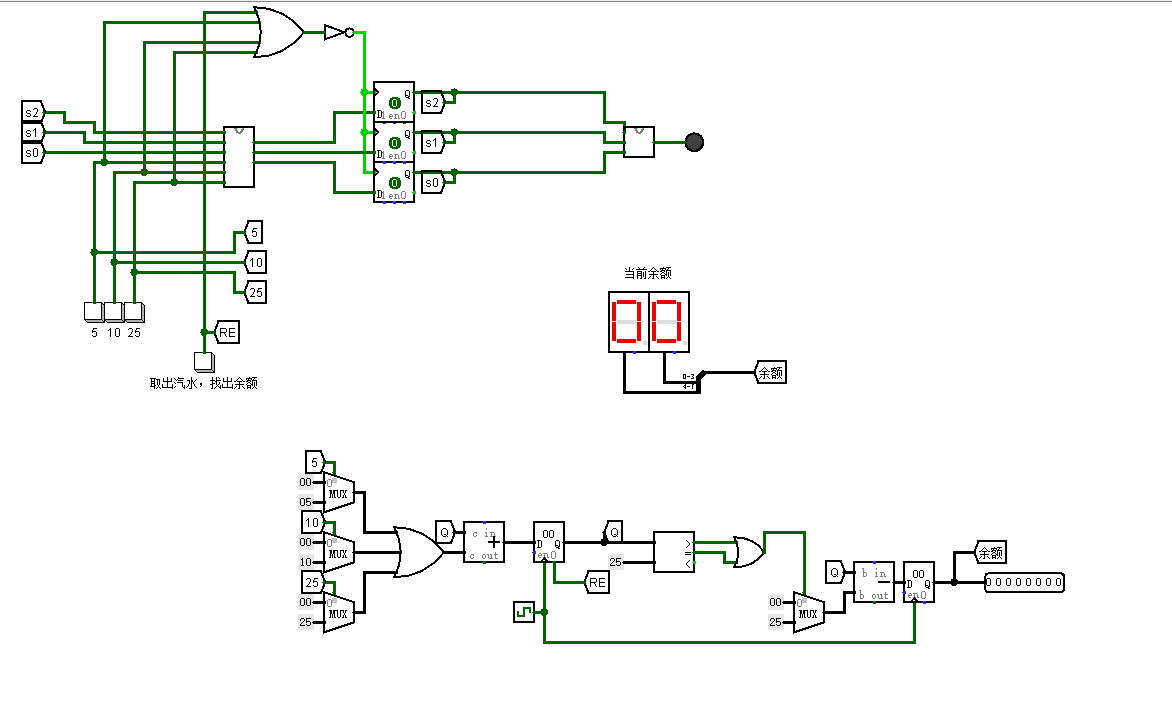
逻辑表达式的推导



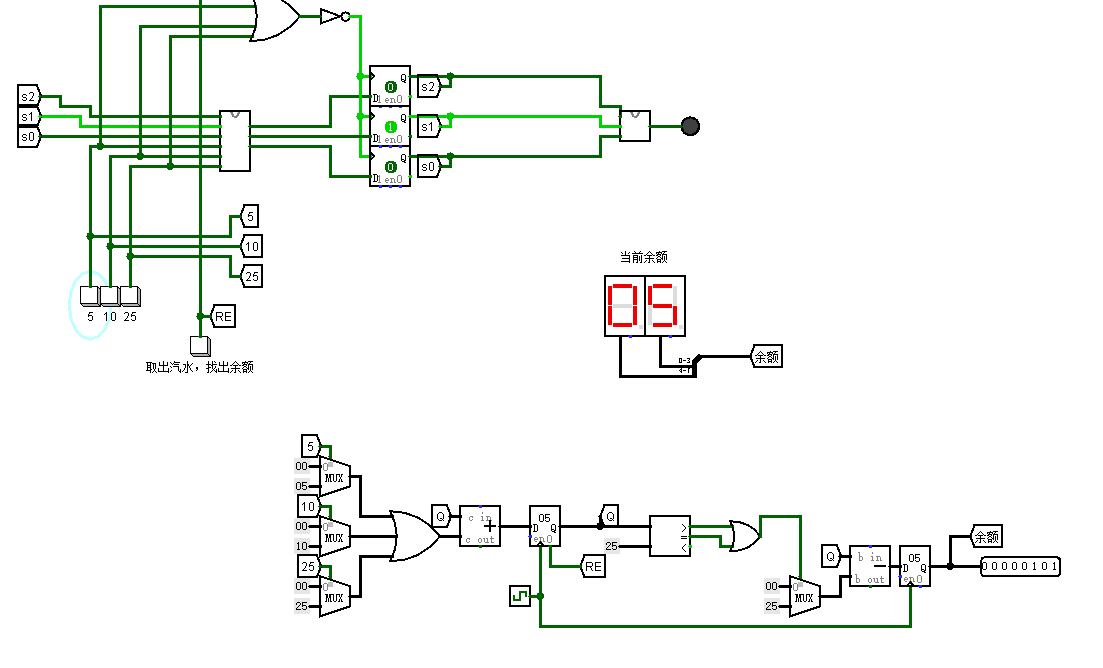
0

有限状态机：

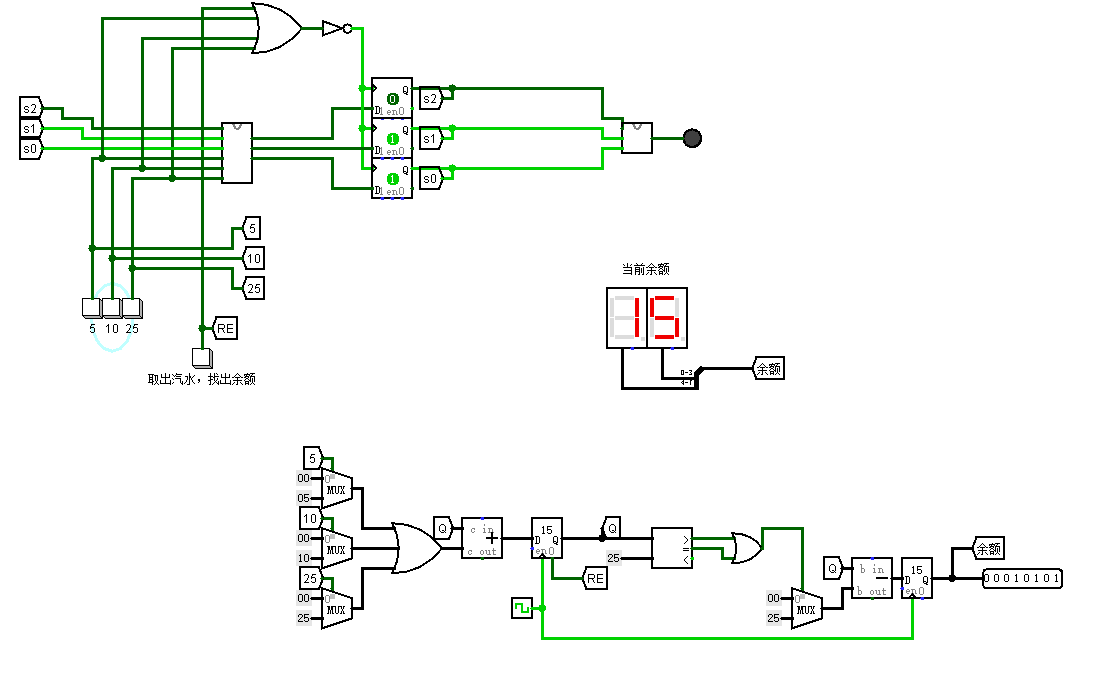




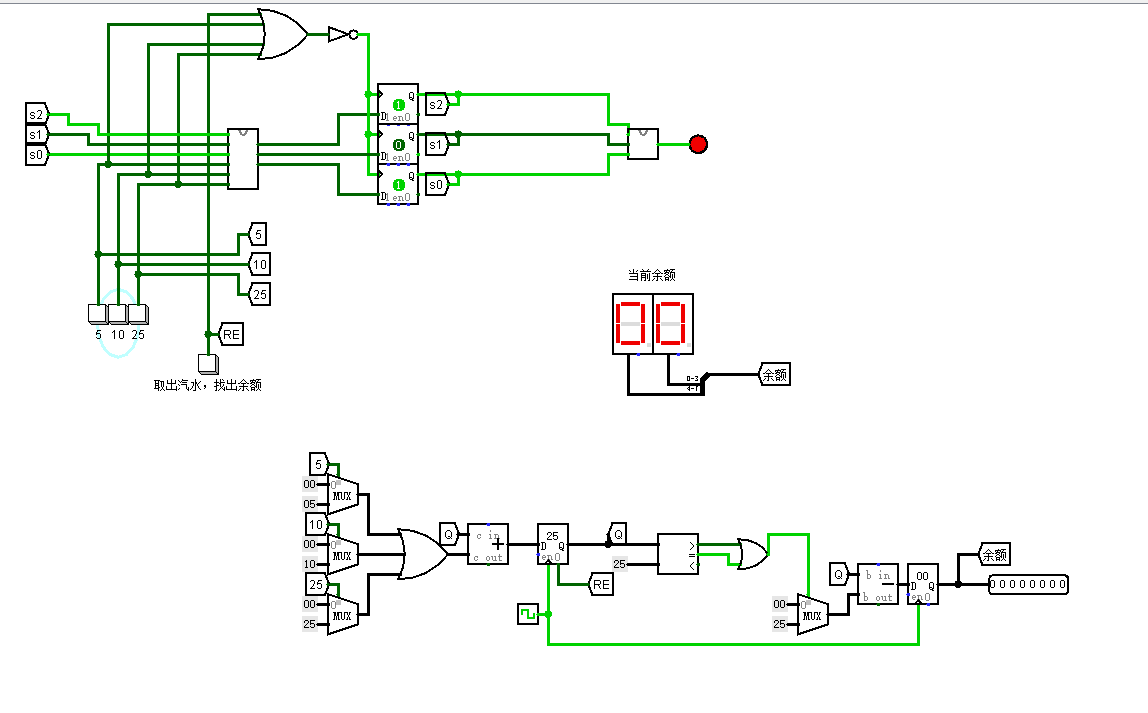
输入5美分时：



输入10美分时：



输入10美分时：



5+10+10==25

输出汽水，余额为零。

# 四 实验细节

应先找零才能进行下一步投钱,即按下re,才能进行下一步。

利用隧道来传递参量，简化电路的连线。

利用有限状态机来实现这个功能，应注意转化表的读取和简化。

# 五 实验总结

通过这个实验，我深化实现了有限状态机的使用。和循环加法器的使用。

# 实验二.附加题

# 一 题目要求

在我们过去的几个实验当中，你已经可以在PYNQ-Z2开发板上用7段数码管显示出计时器的数值。在生活中，我们往往希望用7段数码管显示出更丰富的效果，以达到区分不同运行模式的目的。比如，在设置定时器时，我们希望7段数码管的数字是闪烁的，而完成定时器设置后开始正常计时时，我们希望数字不再闪烁。  
 在这个任务中，你将利用PYNQ-Z2上的按钮切换显示模式（比如长按可以在两种模式之间来回切换），根据显示模式的不同展示不同的显示效果。可供参考的显示效果举例如下：

某个特定的模式下，数字可以在两个七段数码管之间来回切换；

某个特定的模式下，七段数码管的数字可以是闪烁的；

某个特定的模式下，数字可以在LED灯和七段数码管之间来回切换......

# 二 设计思路

利用button控制模式转化，两个按钮控制闪烁和复原。

# 三 成果展现

代码部分：

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 2022/05/03 23:16:38

// Design Name:

// Module Name: led

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module led(

input sw00,sw01,

input clk,

input res1,res2,

output reg[6:0] segments,

output out,

output reg[3:0] led

);

    reg[3:0] data;

    reg[31:0]counter;

    reg shangshuo;

    assign out=sw00;

    always@(posedge clk or negedge res1)

    begin

    if(res2==1'b1)

    shangshuo<=1'b1;

    else

    begin

    if(counter==31250000)

    shangshuo<=1'b1;

    else if(counter==62500000)

    shangshuo<=1'b0;

    else if(counter==93750000)

    shangshuo<=1'b1;

    else if(counter==125000000)

    shangshuo<=1'b1;

    end

    if(res1==1'b1)

    begin

    counter<=0;

    data<=0;

    end

    else begin

    counter<=(counter==125000000)?0:counter+1;

    if(counter==125000000)

    begin

    if(data==9)

    data<=0;

    else

    data<=data+1;

    end

    end

    end

    always@(posedge clk)

    begin

    if(shangshuo)

    begin

    if(sw01)

    begin

    led=4'b0000;

        case(data)

        'h0:segments=7'b111\_1110;

        'h1:segments=7'b011\_0000;

        'h2:segments=7'b110\_1101;

        'h3:segments=7'b111\_1001;

        'h4:segments=7'b011\_0011;

        'h5:segments=7'b101\_1011;

        'h6:segments=7'b101\_1111;

        'h7:segments=7'b111\_0000;

        'h8:segments=7'b111\_1111;

        'h9:segments=7'b111\_1011;

        'ha:segments=7'b111\_0111;

        'hb:segments=7'b001\_1111;

        'hc:segments=7'b000\_1101;

        'hd:segments=7'b011\_1101;

        'he:segments=7'b100\_1111;

        'hf:segments=7'b100\_0111;

        default:segments=7'b000\_0000;

    endcase

    end

    else begin

    segments=7'b000\_0000;

    led=data;

    end

    end

    else begin

    segments=7'b000\_0000;

    led=4'b0000;

    end

    end

endmodule

**实践部分：**

在视频中，闪烁不易通过图片表示出来。

# 四 实验细节

闪烁分为显示数字和断开显示，可以使用时间不同让显示的状态也不同，及1秒亮数字，1秒黑屏。

转换模式用if和button的不同来切换。

# 五 实验总结

我学到了把logisim中的不同状态在vivado上表示，并让他实现可视化在FPGA开发板上。

也写到利用不同的可触按键来实现不同要求。