

# 数逻实验报告 Lab7

雷远航

October 28, 2022

## Abstract

实验项目: 多路选择器设计与应用

## 1 操作方法与实验步骤

### 1.1 数字选择器设计

实验原理图绘制

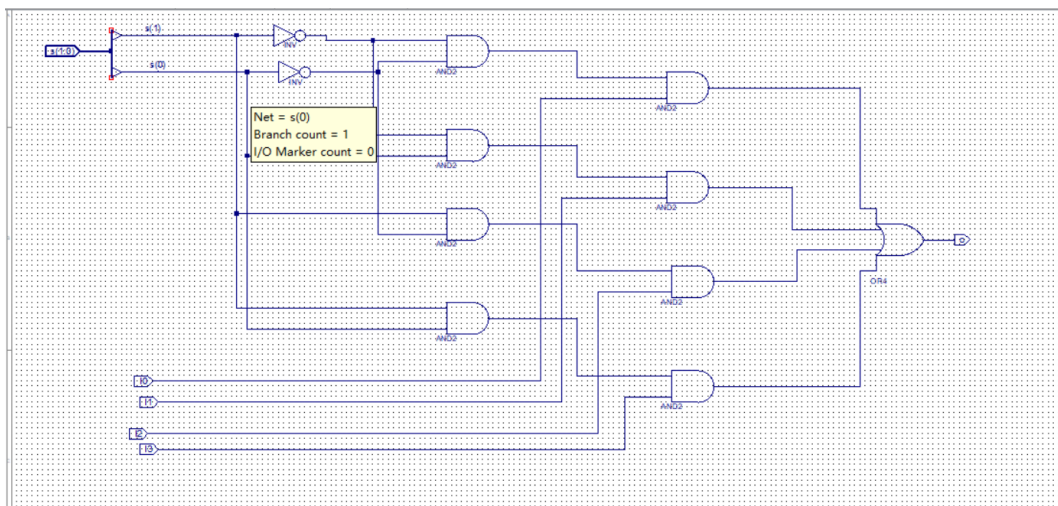


Figure 1: 一位 MUX 绘制

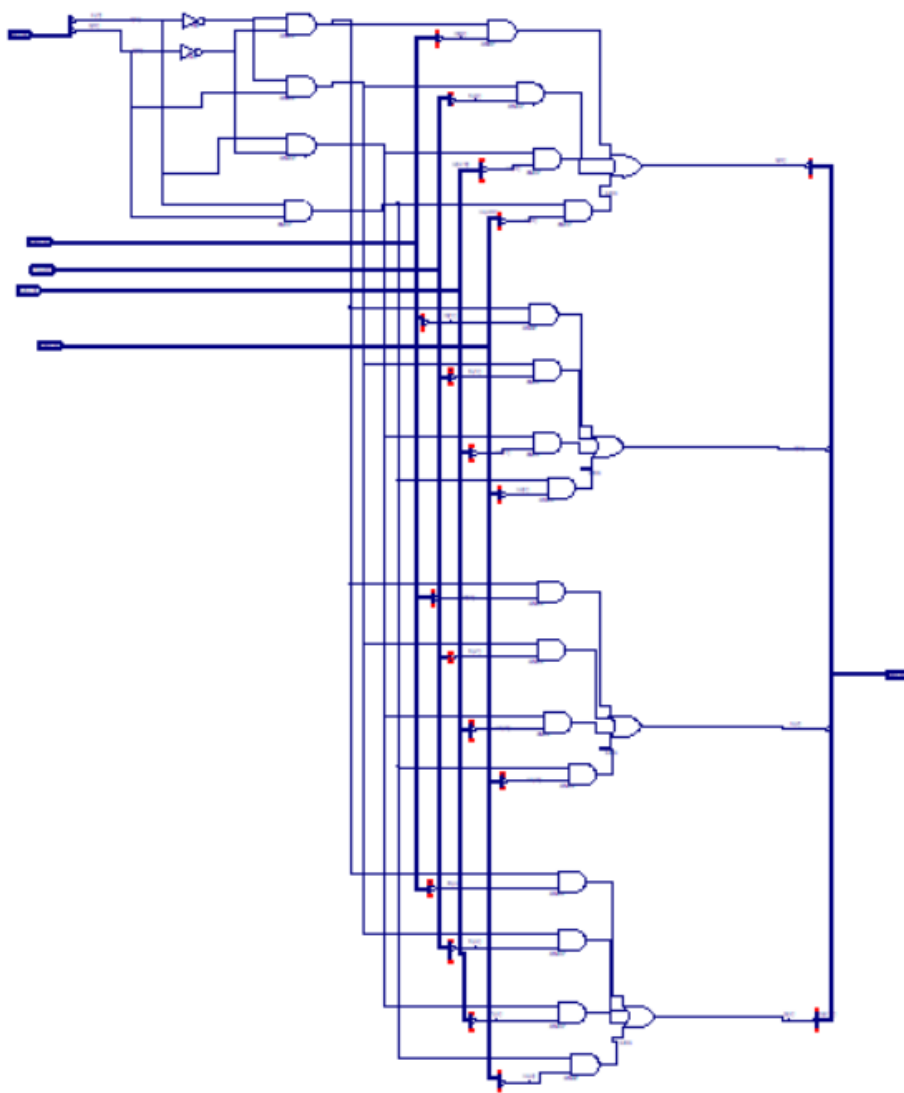


Figure 2: Mux4to1b4

进行模拟仿真

仿真激励代码如下

```

1  `timescale 1ns / 1ps
2
3  module Mux_tb();
4
5      reg[3:0] I0;
6      reg[3:0] I1;
7      reg[3:0] I2;
8      reg[3:0] I3;
9      reg[1:0] s;

```

```

10  wire[3:0] o;
11
12  MUX m0(.I0(I0), .I1(I1), .I2(I2), .I3(I3), .s(s), .o(o));
13
14
15  integer i;
16  initial begin
17      #50
18      #5 I0 = 4'b1100;
19      I1 = 4'b0011;
20      I2 = 4'b1010;
21      I3 = 4'b1111;
22      #5
23      for(i=0;i<=3;i=i+1) begin
24          {s[1:1],s[0:0]} <= i;
25          #50;
26      end
27
28      #50 I0 = 4'b0001;
29      I1 = 4'b0001;
30      I2 = 4'b0100;
31      I3 = 4'b1000;
32      #5
33
34      for(i=0;i<=3;i=i+1) begin
35          {s[1:1],s[0:0]} <= i;
36          #50;
37      end
38
39
40      end
41
42  endmodule

```

## 1.2 记分板设计

相关实验原理图绘制

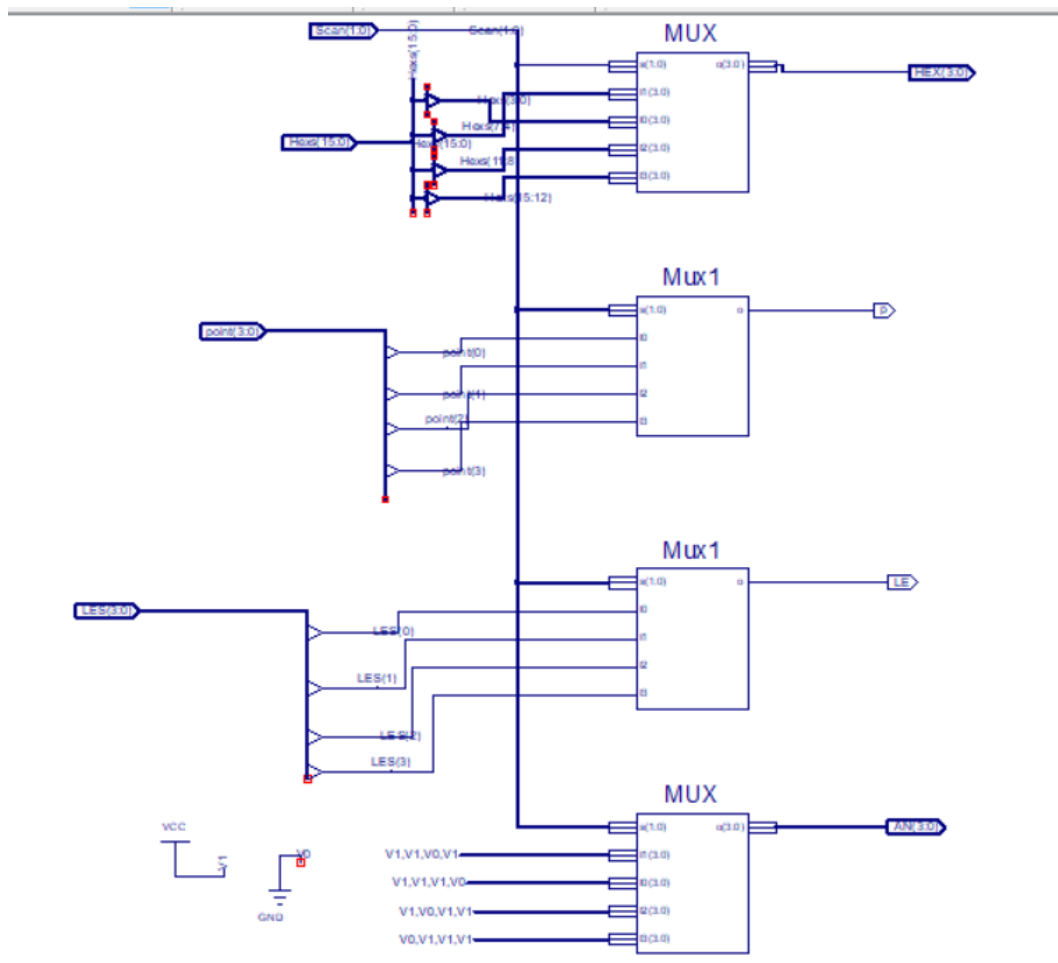


Figure 3: DisPlaySync 原理图绘制

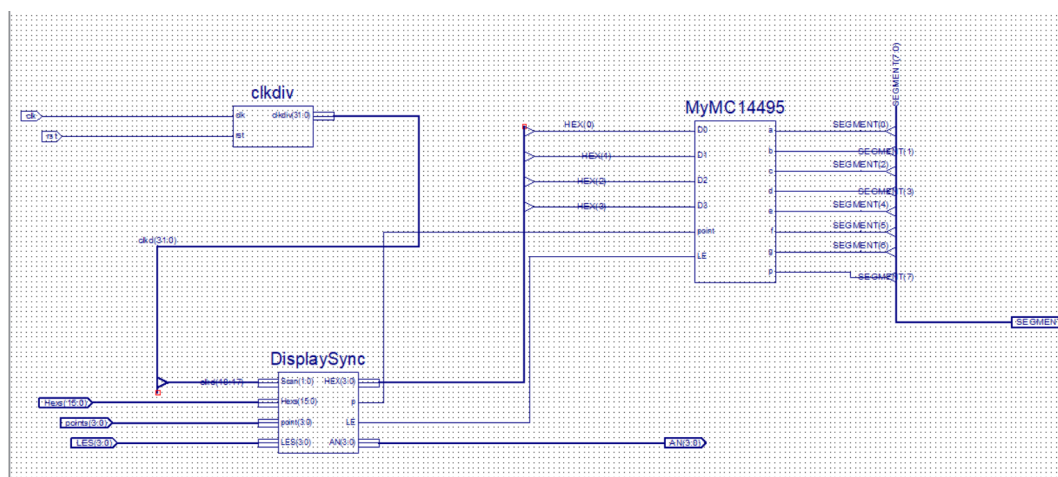


Figure 4: ScoreBoard 原理图绘制

其余实验步骤

导入 clk.v 和 clk\_tb.v, 进行模拟仿真导入 CreateNumber.v 和 CreateNumber\_tb.v, 进行模拟仿真导入 top.v, 形成完整的 ScoreBoard, 进行下版验证.

2 实验结果与分析

2.1 数字选择器设计

仿真波形图

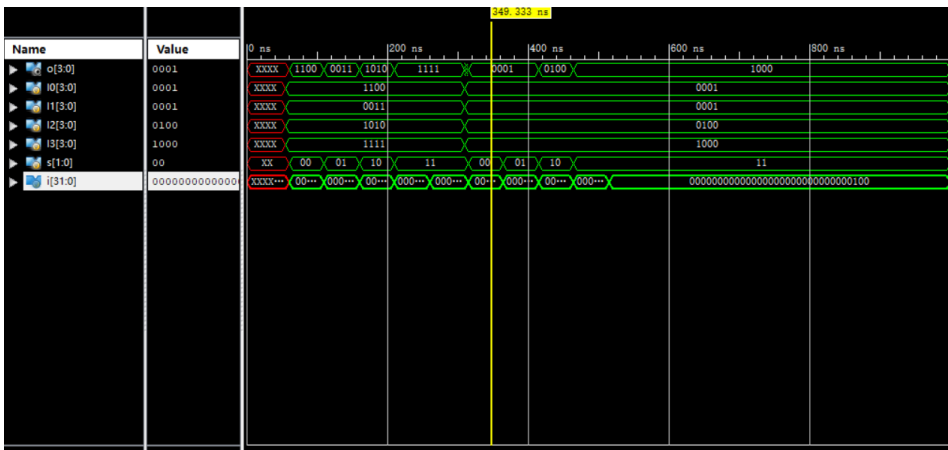


Figure 5: Mux 波形图

波形图解释

在自行书写的仿真激励代码中, 第一轮对选择信号赋初值, 然后通过循环使控制选择的信号从 0-3(二进制形式) 进行遍历, 可以看到,MUX 按照控制信号, 对输出的信号做出了正确的输出, 可以验证所做原理图的正确性. 第二轮循环只是更改了相应的赋值, 实际的过程与第一轮是类似的, 可以看出最终输出的波形仍是正确的.

## 2.2 记分板设计

### 2.2.1 clkdiv 的波形图



Figure 6: clkdiv 波形图

### clkdiv 波形图解释

根据 clkdiv.v 的代码实现可以看出, 当 rst 信号处于时钟上升沿的时候, 输出信号 clkdiv 才会发生变化, 波形图与其相符合. 当 clk 的信号处于时钟上升沿时, clkdiv 所对应的二进制数值会加 1, 在波形图中可以看到, 信号对应的值从 0 开始依次递增与预期相符合. 同时很容易的发现, 第  $k$  个对应的信号变化的周期为  $2^{k+1}$  个单位长度.

### 2.2.2 CreateNumber 的波形图



Figure 7: CreateNumber 波形图

## CreateNumber 波形图解释

根据 CreateNumber.v 可知, num[15:0] 从高到低的 4 位, 依次由 btn[i] 进行控制, 若 btn[i] 的信号为上升边沿 (信号为 1), 则对应位的数值加一. 测试文件中首先对 num 赋初始值为 16'b1010\_1011\_1100\_1101 在 CreateNumber\_tb.v 中, 首先通过控制信号依次使一个, 对应的四位数值加 1, 可以看到对应的数每次在相应的过程中值增加了 1 然后设置 btn 信号为 4'b0000, 可以看到 num 的数值并不发生变化. 最后一次测试 btn 为 4'b1010, 因此第一个 4bits 数和第三个 4bits 数都会发生改变, 波形图中的结果相符.

### 2.2.3 ScoreBoard 下版验证与分析

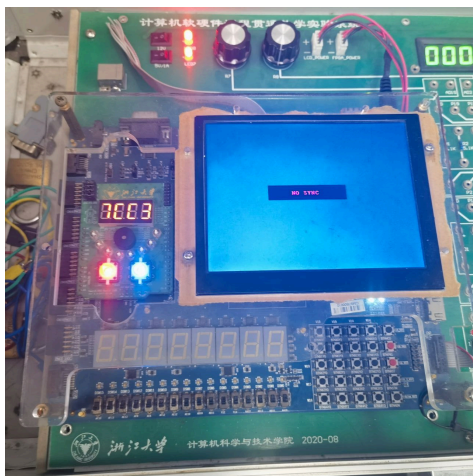


Figure 8: ScoreBoard 下版验证

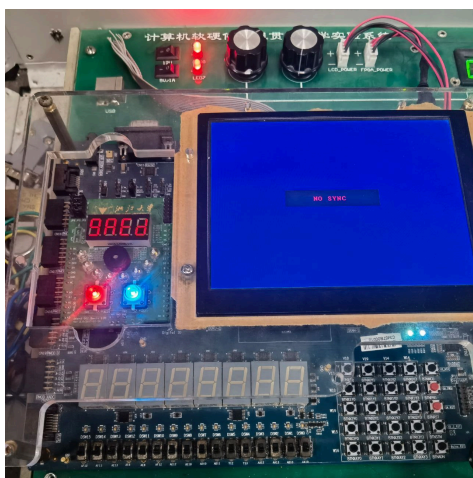


Figure 9: ScoreBoard 下版验证

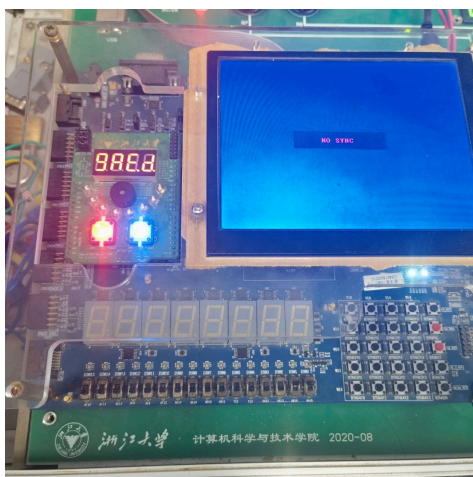


Figure 10: ScoreBoard 下版验证

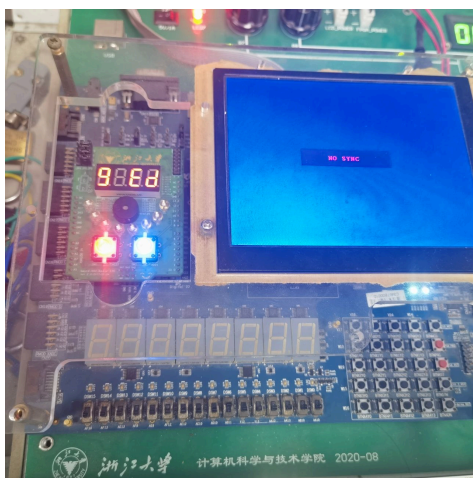


Figure 11: ScoreBoard 下版验证

## 分析

异步显示数字的原理:clkdiv 模块每次时钟信号处于上升沿时,对应的 num 信号递增 1, 实验 中选取 num 的 (16:17) 为作为 DisplaySync 的选择信号, 选择输出一个对应的七段数码管数字, 每次选择输出一个数字, 但由于变化的过程时间间隔过短, 因此人眼看到时会看到四个显示不同的 数字.

数字变化原理: 每当一组数码管对应的按键按下时对应的数字会增加 (由于信号抖动的原因, 增加的数值会大于 1), 当按钮按下时,top.v 中调用的 CreateNumber 模块会使对应的数码管的 数字增加 1, 从而使显示模块显示的数字增加, 但由于没有进行防抖动的处理, 信号检测时会认为信 号的变化不止一次, 因此会造成多次增加.

开关控制原理: 当 LES(3:0) 所对应的四个开关被拨动时, 对应的数字不会显示, 在模块中,LES(3:0) 的四个信号分别对应四组数码管的使能信号, 如果使能信号处于关闭状态时, 那么对应的数码管 便不会显示, 在总的模块中 LES(3:0) 是对应数码管的控制,AN(3:0) 对应的是 clk 信号的控制. points(3:0) 所对应的开关控制了小数点部分的亮暗, 它的控制原理和 LES(3:0) 的控制原理是类似 的.



### 3 讨论与心得

本次实验遇到的主要问题是:(1) 在 DisPlaySync 的原理图绘制时,MUX 对应的四个输入信号在模块中不是按照递增顺序给出的, 在连入对应的输入信号时, 没有看清对应的关系, 导致有两部分对应的信号连接反了, 导致开关控制时, 有两个开关对应的控制数码管弄反, 问题就在于此. (2) 绘制完一位 MUX 之后直接在此基础上绘制了四位 MUX, 没有对一位 MUX 进行保存, 而后续实验又再次用到了一位 MUX, 重新进行的重复的绘制.