浙江大学

本科实验报告

课程名称：数字逻辑设计

姓　　名：颜晗。

学　　院：计算机科学与技术学院

系：

专　　业：计算机科学与技术

学　　号：3200105515

指导教师：蔡铭

2021 年　　11　月　　29　　日

浙江大学实验报告

课程名称：数字逻辑设计　　实验类型：综合

实验项目名称：多路选择器设计及应用

学生姓名：颜晗　专业：计算机科学与技术　学号：3200105515

同组学生姓名：　　指导老师：蔡铭

实验地点：东四 509　　实验日期：2021 年 11 月 8 日

一、实验目的和要求

1.掌握数据选择器的工作原理和逻辑功能

2.掌握数据选择器的使用方法

3.掌握 4 位数码管扫描显示方法

4.4 位数码管显示应用—记分板设计

二、实验内容和原理

实验内容：

任务 1：数据选择器设计；

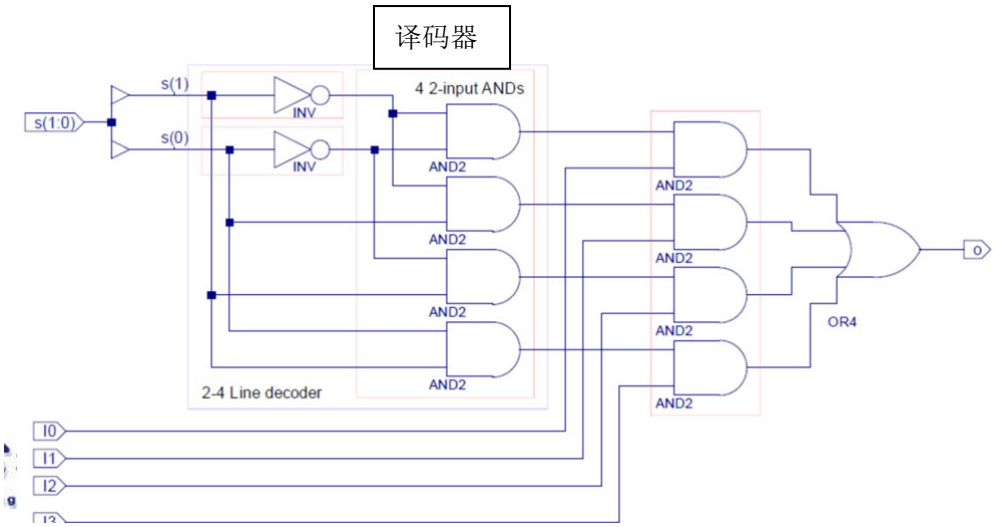
任务 2：记分板设计。

实验原理：

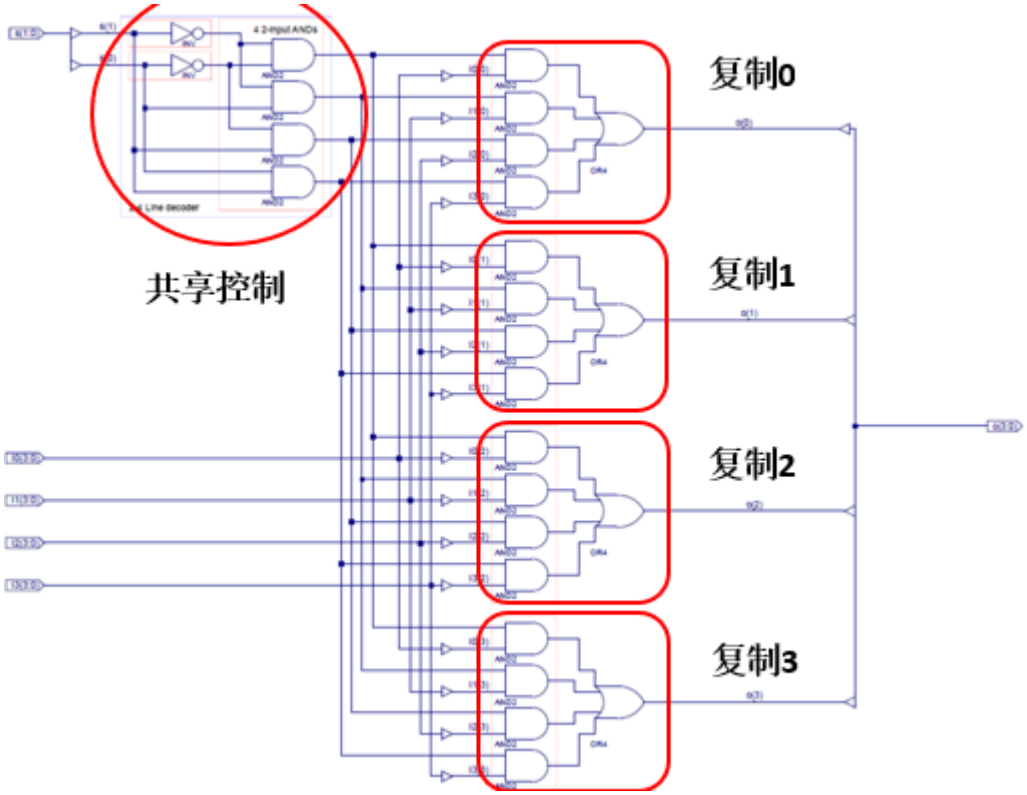
1. 四选一多路选择器：MUX4to1 和 4 位四选一扩展：MUX4to1b4

本质即通过控制信号从四个（组）信号中选择一个（组）进行输出，使用与门和或门控制某一信号的输入输出。

信息输入	控制端	选择输出	
I0 I1 I2 I3	S1 S0	o	输出项
I0 I1 I2 I3	0 0	I0	S1S0 I0
I0 I1 I2 I3	0 1	I1	S1S0 I1
I0 I1 I2 I3	1 0	I2	S1S0 I2
I0 I1 I2 I3	1 1	I3	S1S0 I3



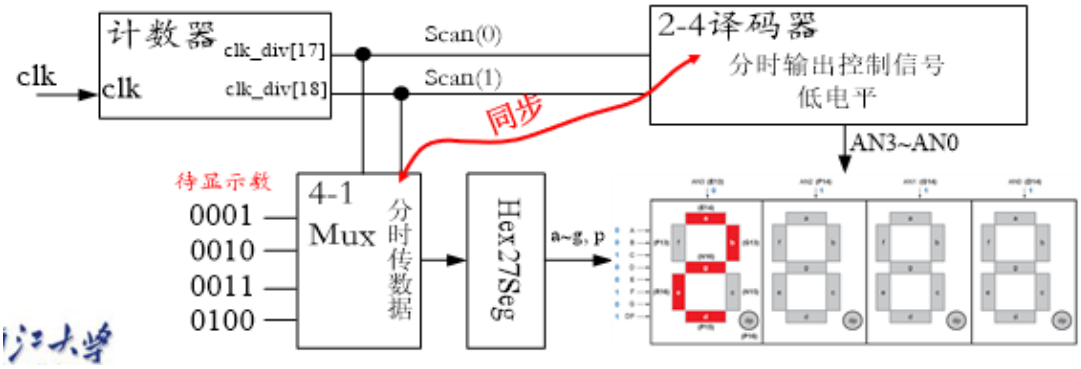
四位的多路选择器原理相同，不过是控制信号需要多分几路以控制一组信号，而同时不同组的信号相同的位通过同一个或门输出。



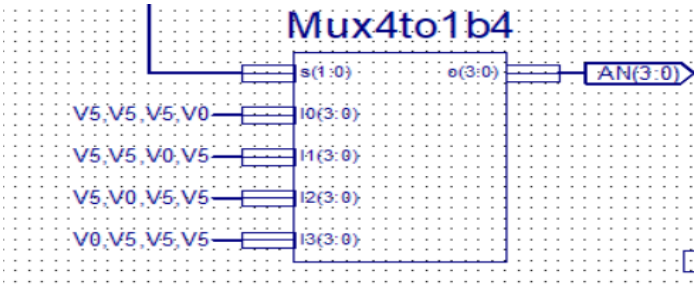
2. 动态扫描显示

2.1 扫描信号来自计数器：将时序转化为组合电路。

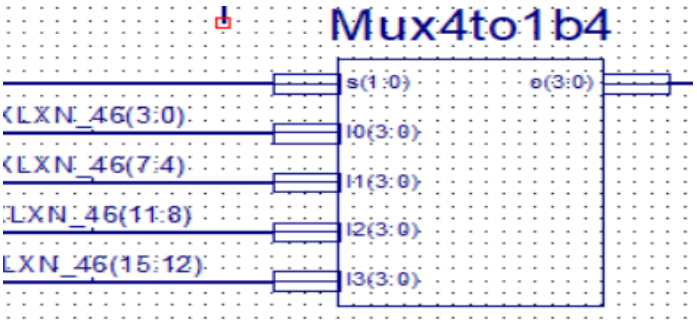
2.2 由板载时钟 clk(100MHz)作为计数器时钟，分频后的高两位信号（clk_div[18:17]）作为扫描控制信号，输入 2-4 译码器控制哪个数码管显示（位选择），同时输入 4 选 1 多路复用器选择需要显示哪个数据（段码选择）



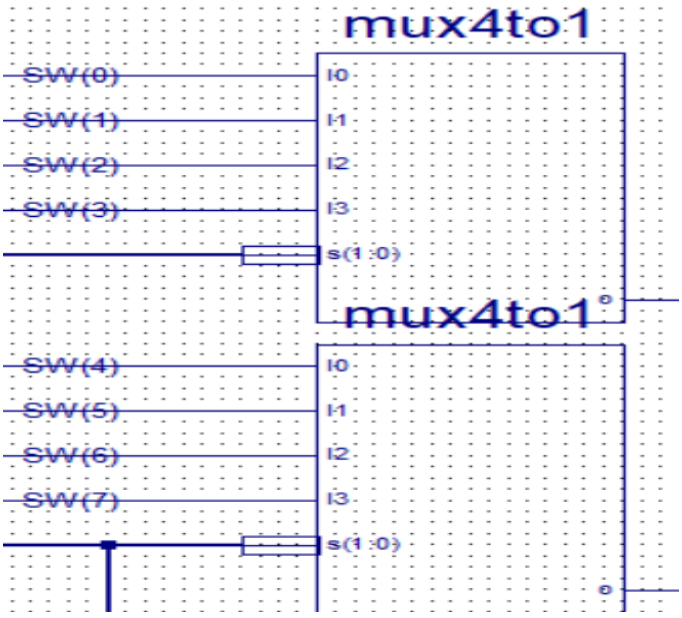
具体实现：我们需要两个四位的多路选择器来分别控制需要输出的数和其对应的数码管显示。另外还需要两个一位的多路选择器控制数和小数点的消隐。



仅有输入 V0 信号的数码管可以显示，以此控制数码管的扫描显示。



四组输入分别对应四个数码管显示的数且显示的数各自独立，不必相同。和数码管由同一信号控制，做到一个数码管对应一个数字。



两个选择器可分别控制点与数字的消隐，即如果这两个的输出控制数字与点无法显示，即使扫描到了对应的数码管也不能显示；反之也是一样的，即只有两个控制消隐的信号都可以显示，数字才能显示出来。但是虽然无法显示，按钮增加依旧生效。

三、实验过程和数据记录

1. 任务一：数据选择器设计

1.1 新建工程和源文件，设计 Mux4to1 和 Mux4to1b4

1.2 Verilog 代码如下，两者基本相同：

```
module Mux4to1(
    input [1:0]s,
    input I0,
    input I1,
    input I2,
    input I3,
    output reg o
    );
   always @(*) begin
      case (s)
      2'h0: o <= I0;
      2'h1: o <= I1;
      2'h2: o <= I2;
      2'h3: o <= I3;
      endcase
   end
endmodule
```

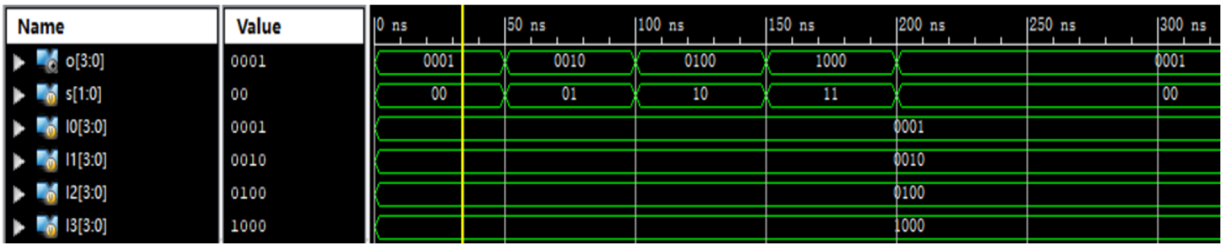
```
module Mux4to1b4(
  input [1:0] s,
  input [3:0] I0,
  input [3:0] I1,
  input [3:0] I2,
  input [3:0] I3,
  output reg [3:0] o
  );
   always @(*) begin
      case(s)
         2'b00:
            o <= I0;
         2'b01:
            o <= I1;
         2'b10:
            o <= I2;
         2'b11:
            o <= I3;
      endcase
   end
endmodule
```

1.3 检查设计，综合，创建模块 Mux4to1 和 Mux4to1b4。

1.4 对 Mux4to1b4 模块进行仿真，仿真代码如下：

```
   initial begin
      // Initialize Inputs
      s = 0;
      I0 = 4'b0001;
      I1 = 4'b0010;
      I2 = 4'b0100;
      I3 = 4'b1000;
      // Wait 100 ns for global

      for(s=0;s<=2;s=s+1)begin
         #50;
      end
      #50;
      s = 0;
      // Add stimulus here

   end

endmodule
```

仿真波形如下，对于不同的控制信号，都可以正确输出对应的输入信号：

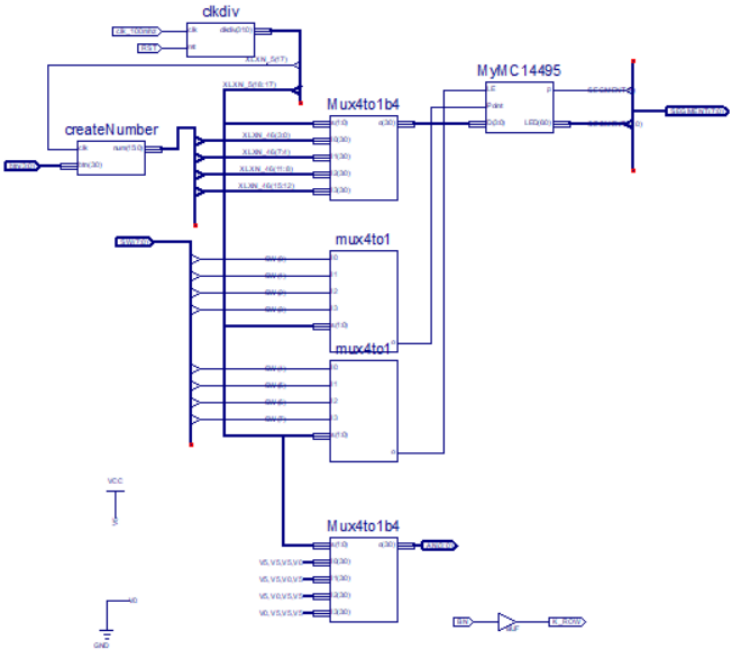


2.任务 2：记分板设计

2.1 新建工程

2.2 将任务一设计的两个选择器和之前的 Mc14495 加入工程。可以分别建立工程设计模块或者直接在该工程中利用 Verilog 代码设计模块“pbdebounce”,“clkdiv”,“createNumber”，利用“Set as Top Module”便可分别对模块进行综合。注意要先设计“pbdebounce”模块才能设计“createNumber”模块。

2.3 再新建源文件 scoreboard 并“Set as Top Module”，利用原理图或 Verilog 代码设计。



原理图过于繁琐可以使用代码：

```
module Scoreboard_ver(
    input clk_100mhz,
    input RST,
    input BN,
    input [3:0] btn,
    input [7:0] SW,
    output K_ROW,
    output [3:0] AN,
    output [7:0] SEGMENT
    );
      //声明变量
   wire [31:0] clkdiv;
   wire V0;
   wire V5;
   wire [15:0] num;
   wire [3:0] out;
   wire LE;
   wire Point;

   //调用模块：模块名 实例名(参数)
   clkdiv f1(.clk(clk_100mhz),.rst(RST),.clkdiv(clkdiv));
   createNumber f2(.btn(btn[3:0]),.clk(clkdiv[17]),.num(num[15:0]));
```

```
Mux4tolb4 f3(.l0(num[3:0]),
             .l1(num[7:4]),
             .l2(num[11:8]),
             .l3(num[15:12]),
             .s(clkdiv[18:17]),
             .o(out[3:0]));
mux4tol f4(.I0(SW[0]),
           .I1(SW[1]),
           .I2(SW[2]),
           .I3(SW[3]),
           .s(clkdiv[18:17]),
           .o(Point));
mux4tol f5(.I0(SW[4]),
           .I1(SW[5]),
           .I2(SW[6]),
           .I3(SW[7]),
           .s(clkdiv[18:17]),
           .o(LE));
Mux4tolb4 f6(.l0({V5, V5, V5, V0}),
             .l1({V5, V5, V0, V5}),
             .l2({V5, V0, V5, V5}),
             .l3({V0, V5, V5, V5}),
             .s(clkdiv[18:17]),
             .o(AN[3:0]));
```

2

```
Mc14495 f7( .D0(out[0]),
            .D1(out[1]),
            .D2(out[2]),
            .D3(out[3]),
            .LE(LE),
            .point(Point),
            .a(SEGMENT[0]),
            .b(SEGMENT[1]),
            .c(SEGMENT[2]),
            .d(SEGMENT[3]),
            .e(SEGMENT[4]),
            .f(SEGMENT[5]),
            .g(SEGMENT[6]),
            .p(SEGMENT[7]));
```

3

```
BUF f8(.I(BN), .O(K_ROW));
VCC f9(.P(V5));
GND f10(.G(V0));

endmodule
```

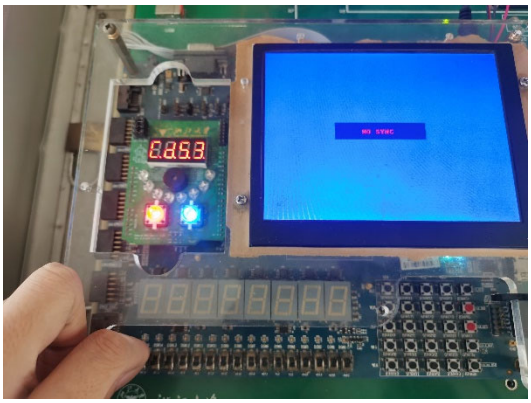
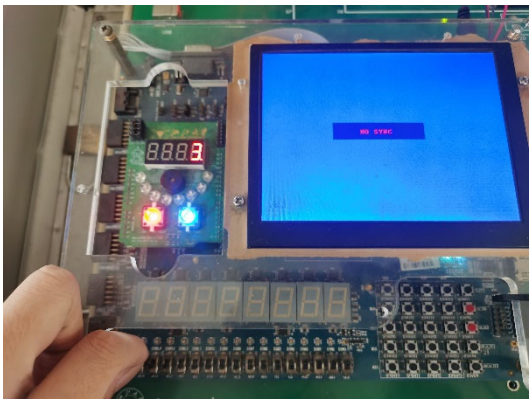
2.4 建立引脚约束文件

```
NET "clk_100mhz"        LOC = AC18      | IOSTANDARD = LVCMOS18;
NET "RST"               LOC = AF10      | IOSTANDARD = LVCMOS15;
NET "SW[0]"             LOC = AA10      | IOSTANDARD = LVCMOS15;
NET "SW[1]"             LOC = AB10      | IOSTANDARD = LVCMOS15;
NET "SW[2]"             LOC = AA13      | IOSTANDARD = LVCMOS15;
NET "SW[3]"             LOC = AA12      | IOSTANDARD = LVCMOS15;
NET "SW[4]"             LOC = Y13       | IOSTANDARD = LVCMOS15;
NET "SW[5]"             LOC = Y12       | IOSTANDARD = LVCMOS15;
NET "SW[6]"             LOC = AD11      | IOSTANDARD = LVCMOS15;
NET "SW[7]"             LOC = AD10      | IOSTANDARD = LVCMOS15;
NET "SEGMENT[0]"        LOC = AB22      | IOSTANDARD = LVCMOS33 ;
NET "SEGMENT[1]"        LOC = AD24      | IOSTANDARD = LVCMOS33 ;
NET "SEGMENT[2]"        LOC = AD23      | IOSTANDARD = LVCMOS33 ;
NET "SEGMENT[3]"        LOC = Y21       | IOSTANDARD = LVCMOS33 ;
NET "SEGMENT[4]"        LOC = W20       | IOSTANDARD = LVCMOS33 ;
NET "SEGMENT[5]"        LOC = AC24      | IOSTANDARD = LVCMOS33 ;
NET "SEGMENT[6]"        LOC = AC23      | IOSTANDARD = LVCMOS33 ;
NET "SEGMENT[7]"        LOC = AA22      | IOSTANDARD = LVCMOS33 ;
NET "AN[3]"             LOC = AC22      | IOSTANDARD = LVCMOS33 ;
NET "AN[2]"             LOC = AB21      | IOSTANDARD = LVCMOS33 ;
NET "AN[1]"             LOC = AC21      | IOSTANDARD = LVCMOS33 ;
NET "AN[0]"             LOC = AD21      | IOSTANDARD = LVCMOS33 ;

NET "btn[3]"            LOC = V18       | IOSTANDARD = LVCMOS18;
NET "btn[2]"            LOC = V19       | IOSTANDARD = LVCMOS18 ;
NET "btn[1]"            LOC = V14       | IOSTANDARD = LVCMOS18 ;
NET "btn[0]"            LOC = W14       | IOSTANDARD = LVCMOS18 ;

NET "K_ROW"             LOC = V17       | IOSTANDARD = LVCMOS18 ;
NET "BN"                LOC = AF13      | IOSTANDARD = LVCMOS15 ;
```

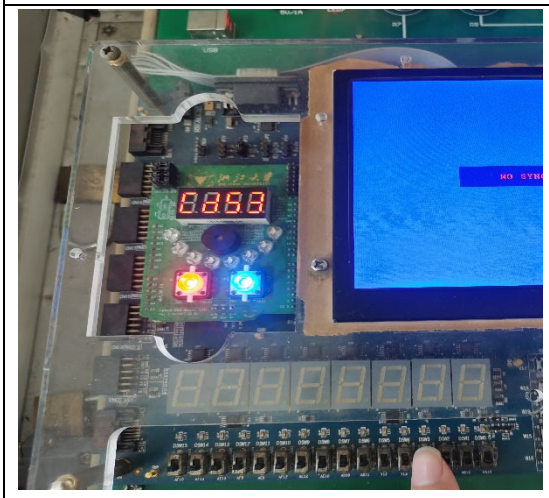
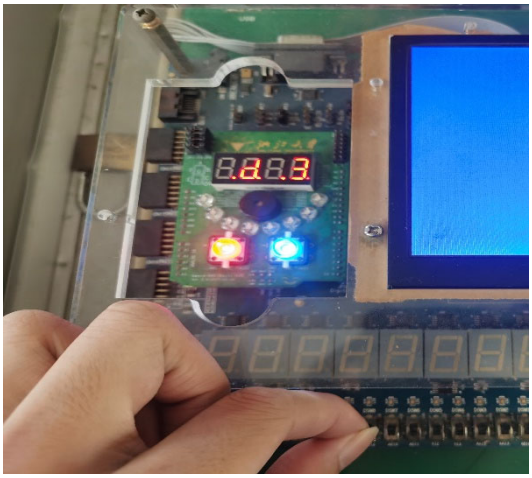
2.5 检查错误，综合，实现设计，上板检查。

四、实验结果分析

1. 复位信号，拨动开关使晶体管显示固定，无法改变

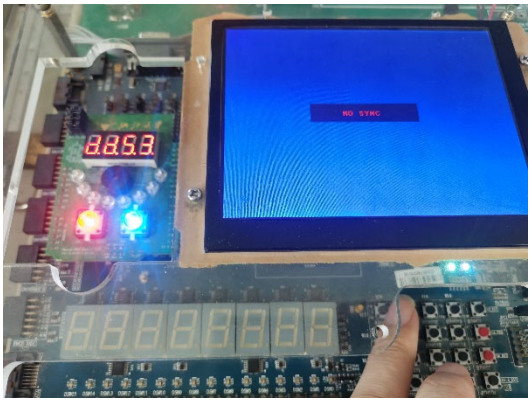
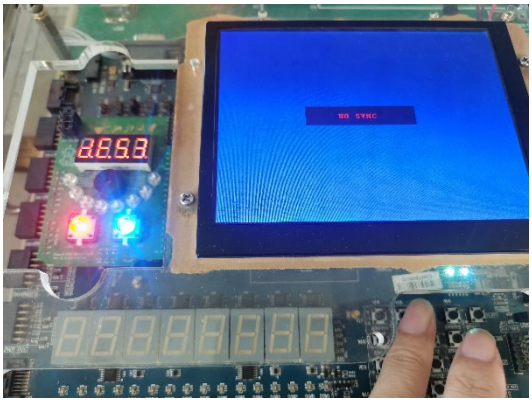



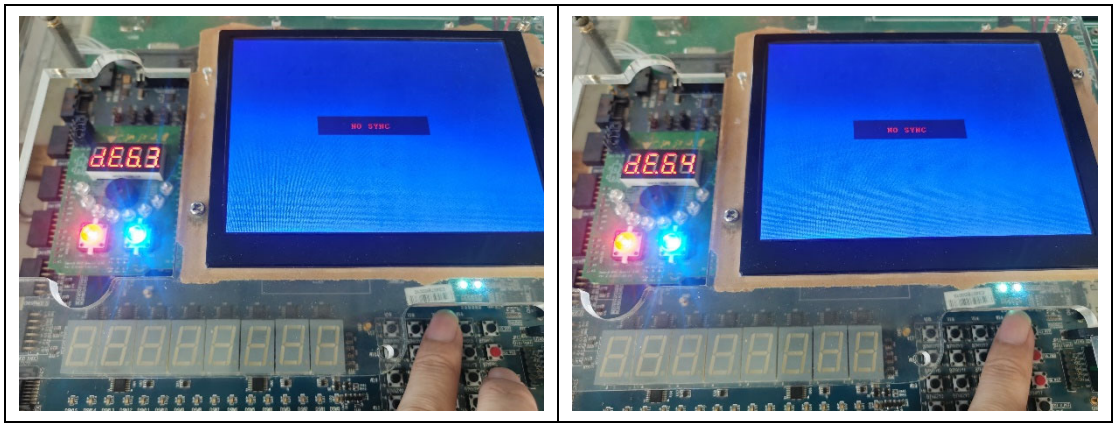
2. 点与数字的消隐控制

点消失	数字消失
	

3. 按动按钮，逐一改变个各晶体管显示数字，各自独立，不受影响。

（原本显示数字 cd53，逐一改变+1 后变为 de64）



五、实验总结与反思

本次实验相比前一次实验增加了许多模块，虽然部分模块直接由老师给出，但是一时间无法理解原理也给实验带来了一定的困难，而且大量的模块也给排除错误带来了困难，尤其是经验不足的情况下无法确认出错的模块；本次实验中最大的错误在 Mux4to1b4 模块的设计错误，本来是 I0~3 分别为四组信号，实验时却由于未看清原理图，嫌繁琐用 Verilog 设计时将同一位不同组的四个信号（如 I0~3 的第一位）当成了一组(这样使仿真时的波形与正确的是一致的但是输出的逻辑却是不同的)，直接导致了后面反复寻找错误，浪费了大量时间。带来了深刻的教训，以后一定要分辨清楚模块的逻辑再设计。