浙江大学

本科实验报告

课程名称：	数字逻辑设计
姓　　名：	蒋奕
学　　院：	计算机学院
系：	计算机系
专　　业：	计算机科学与技术
学　　号：	3210103803
指导教师：	马德

2022 年　10　月　31　日

浙江大学实验报告

课程名称：数字逻辑设计　实验类型：　绘图实验+模块调用

实验项目名称：多路选择器设计及应用

学生姓名：蒋奕　专业：　计算机科学与技术　学号：　3210103803

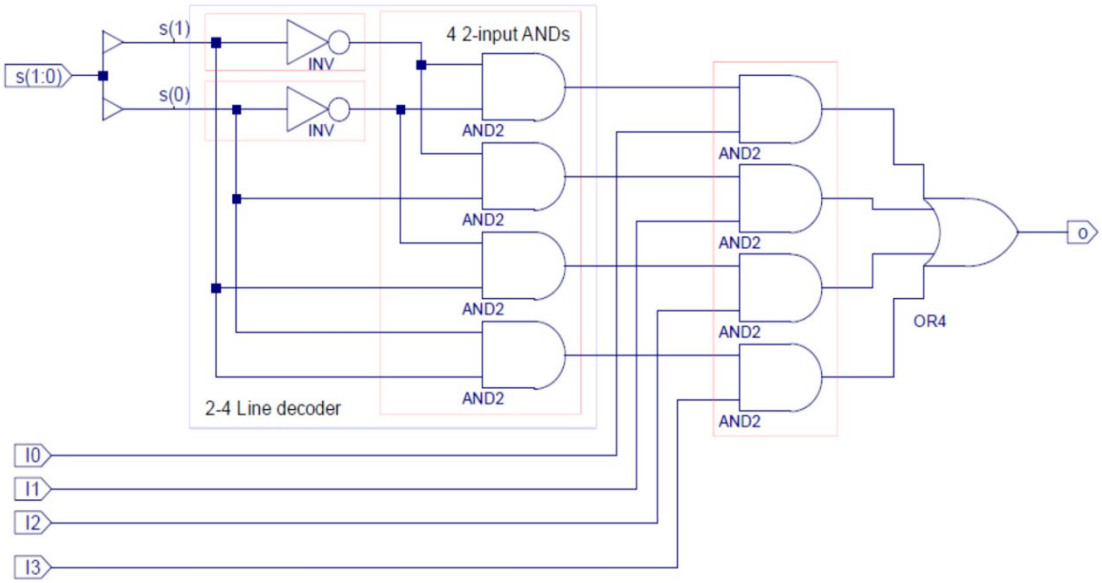
同组学生姓名：　任庭旭　指导老师：马德

实验地点：　东 4-509　实验日期：　2022 年　10 月 31 日

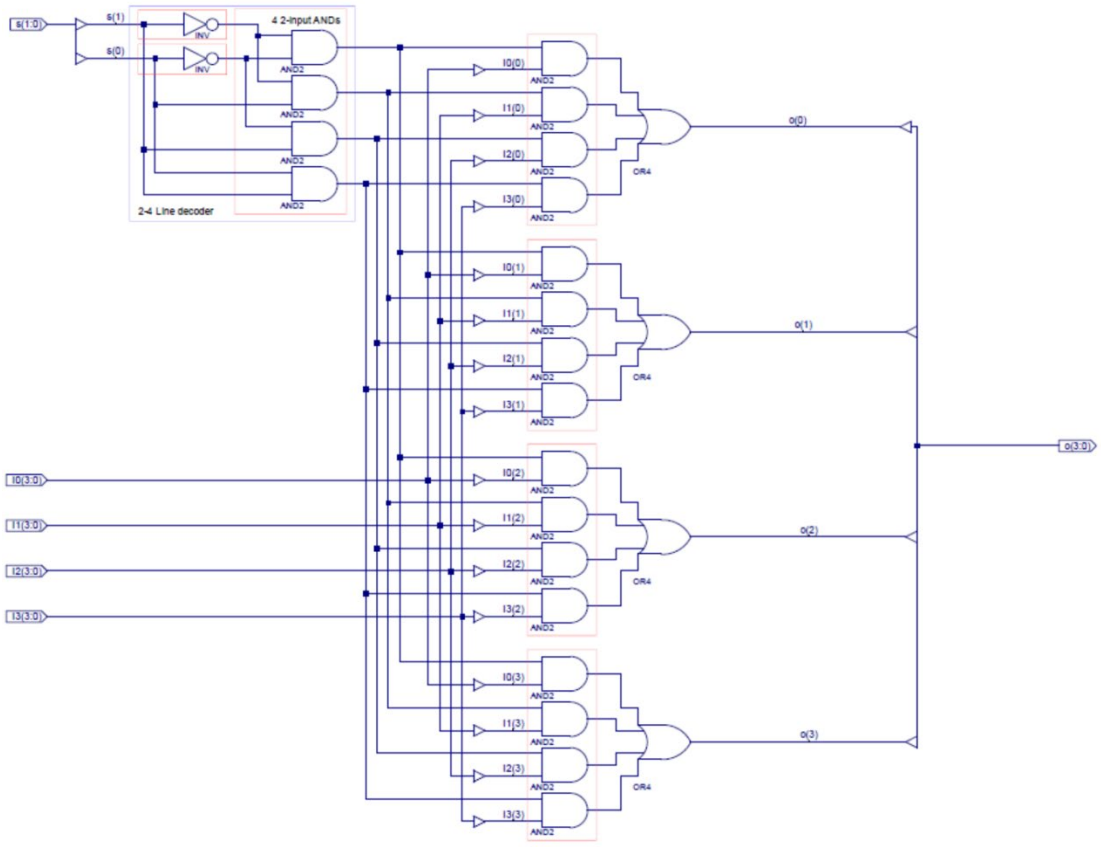
一、　实验目的：

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

实验一：数据选择器设计
新建工程，工程名称用 Mux4to1b4_sch。
新建 Schematic 源文件，文件名称用 Mux4to14b。
用如下原理图进行设计 MUX4to1。



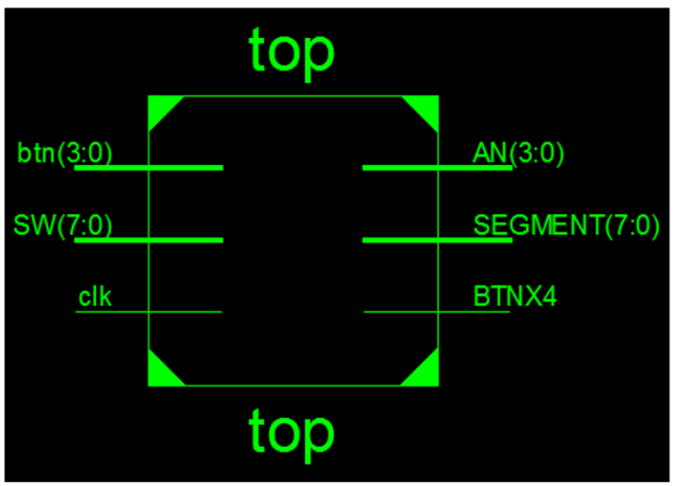
用如下原理图进行设计 MUX4to1b4。



建立仿真波形文件，初始化 I0、I1、I2、I3 和 S[1:0]，进行仿真

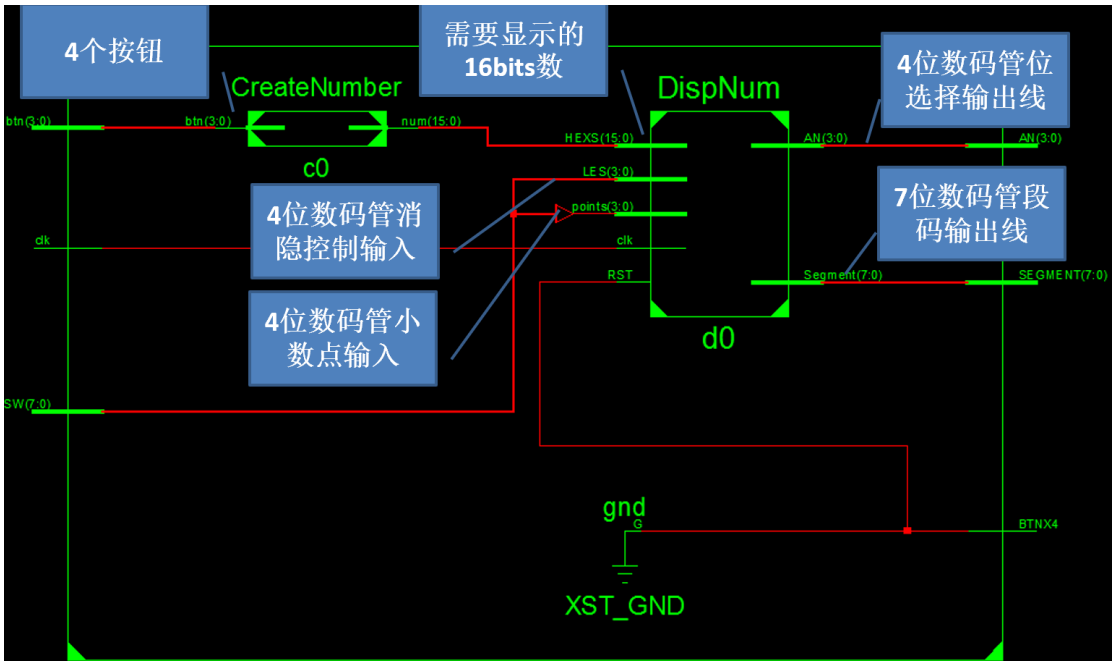
实验二：记分板设计
新建工程 ScoreBoard。Top Level Source Type 用 HDL。
设计动态扫描同步输出模块
通用计数分频模块

顶层模块

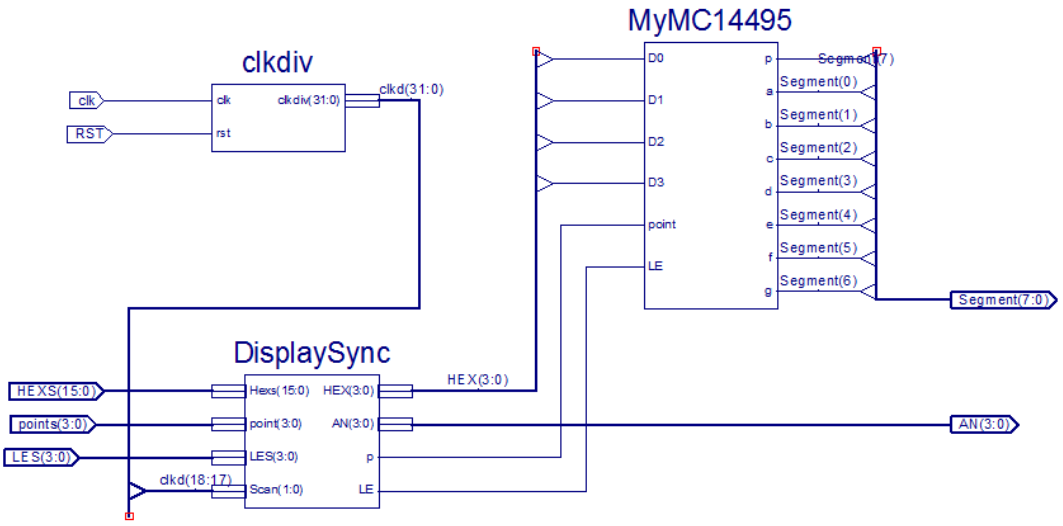


输出输出引脚功能
输入
时钟：clk
使能控制：sw[7:4]
小数点输入：sw[3:0]
按键输入数字：BTNX4Y0-BTNX4Y3 为 btn[3:0]
输出
七段数码管段码输出线：segment[7:0}，包括 a-g，p
七段数码管位选择线：an[3:0]
按键使能控制线：BTNX4
根据设计修改 UCF

其中顶层模块内部结构为



DispNum 模块内部结构如下：



用原理图形式设计 DisplaySync 模块
模块名：DisplaySync.sch
制作逻辑符号并修改：DisplaySync.sym
相关参数如下：
Hexs(15:0)：需要显示的 4 个 4 位二进制数
point(3:0)：每位数码管的小数点
LES(3:0)：每位数码管是否需要消隐
Scan(1:0)：扫描控制信号
HEX(3:0)：当前要显示的 4 位二进制数
AN(3:0)：4 位数码管的位选择信号（低电平有效）
P、LE：小数点和消隐控制

用 Verilog 代码设计辅助模块：时钟计数分频器
32 位时钟计数分频器延时较高，要求不高的时钟也可以用
本实验中用 clkdiv(18:17)作为扫描控制信号，控制 4 位数码管的动态扫描，每一位显示切换时间为 2^17 / 100M = 1.3ms
制作逻辑符号并修改：clkdiv.sym
相关参数如下：
模块名：clkdiv.v
clk：实验板主时钟
rst：复位信号
clkdiv(31:0)：分频时钟输出

设计顶层模块
新建源文件 top，在右键菜单里设为“Top Module”，Verilog 代码如下：

```
module top(input wire clk,
    input wire [7:0] SW,
    input wire [3:0] btn,
    output wire [3:0] AN,
    output wire [7:0] SEGMENT,
    output wire BTNX4
);
    wire [15:0] num;
    CreateNumber c0(btn,num);
    DispNum d0(clk, num, SW[7:4], SW[3:0], 1'b0, AN, SEGMENT);
     assign BTNX4 = 1'b0; //Enable button inputs
endmodule
```

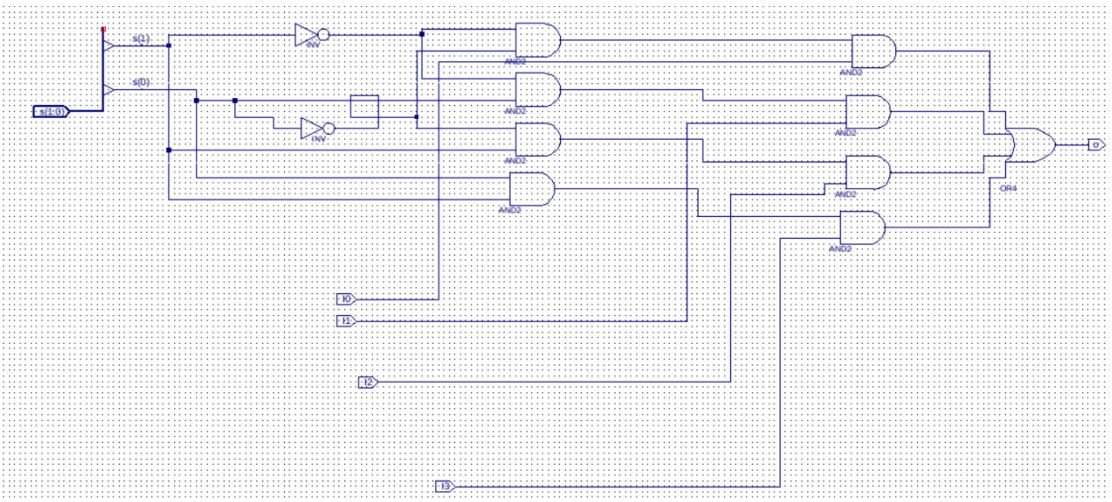
设计 CreateNumber 按键数据输入模块，Verilog 代码如下：

```
module CreateNumber(
       input wire [3:0] btn,
       output reg [15:0] num
    );
wire [3:0] A,B,C,D;
initial num <= 16'b1010_1011_1100_1101;
assign A = num[3:0] + 4'd1;
assign B = num[7:4] + 4'd1;
assign C = num[11:8] + 4'd1;
assign D = num[15:12] + 4'd1;
always @(posedge btn[0]) num[3:0] <= A;
always @(posedge btn[1]) num[7:4] <= B;
always @(posedge btn[2]) num[11:8] <= C;
always @(posedge btn[3]) num[15:12] <= D;
endmodule
```

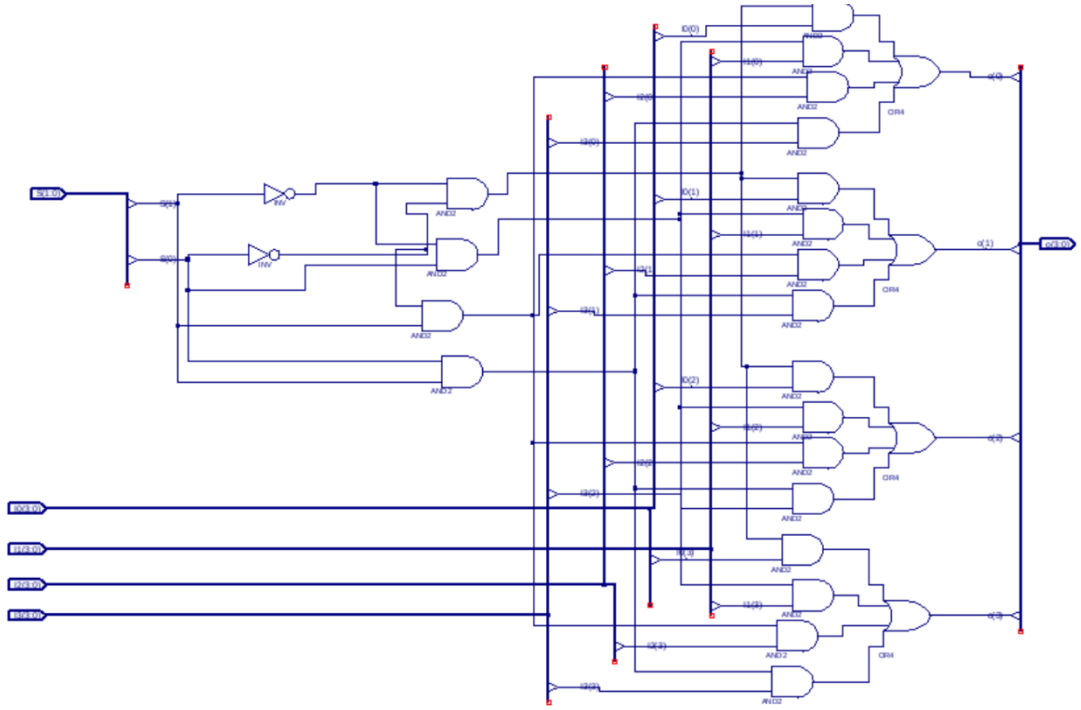
之后建立 k7.ucf 进行验证

二、实验数据记录和处理

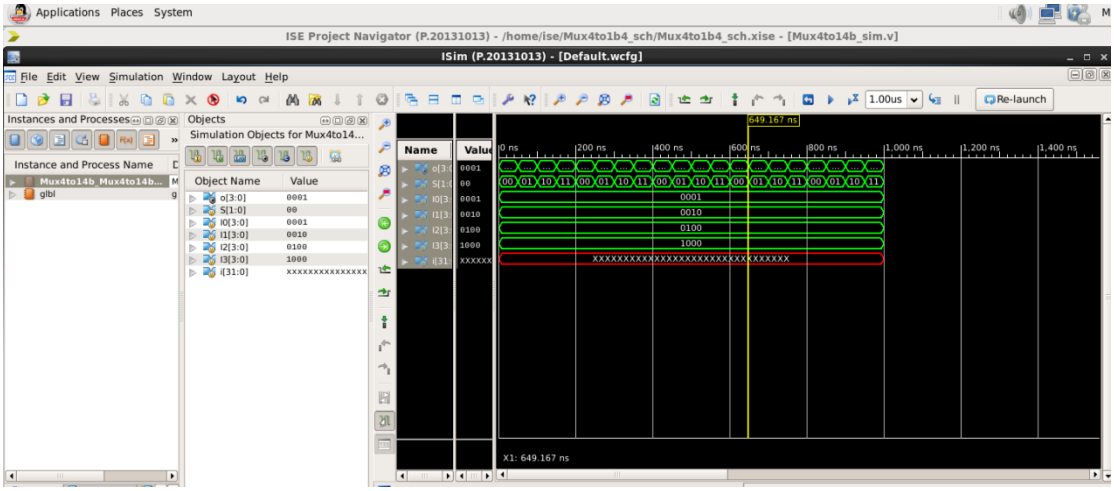
实验一：数据选择器设计



MUX4to1 自主作图如上



MUX4to14b 自主作图如上



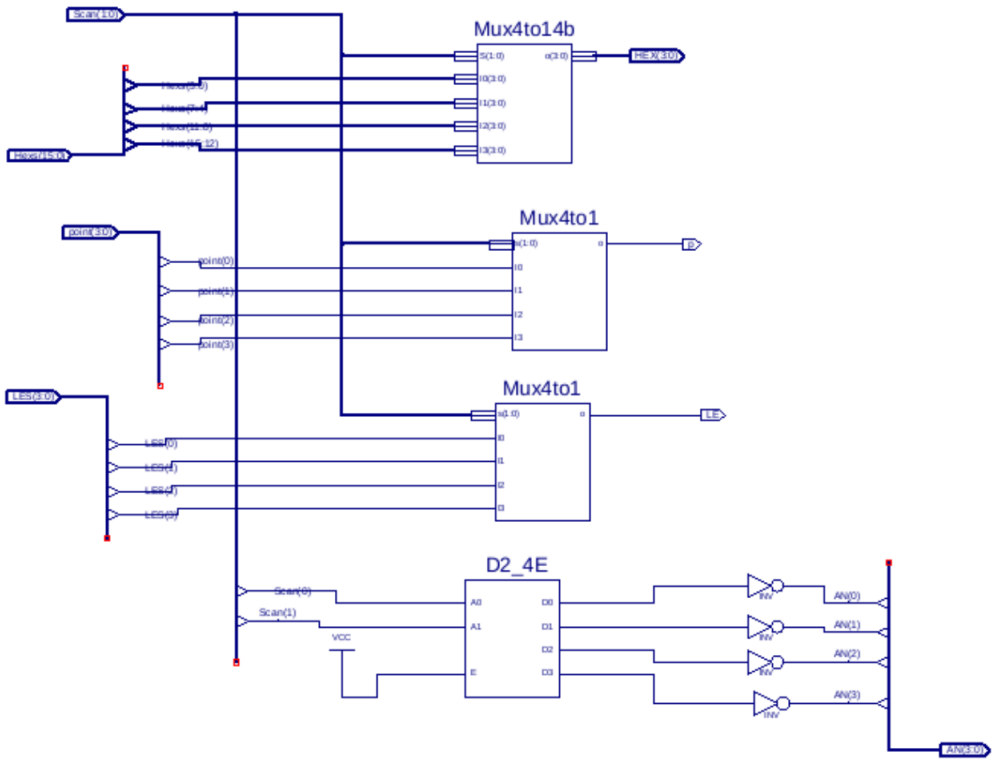
实验仿真波形如上

对应的 Verilog 仿真代码如下：

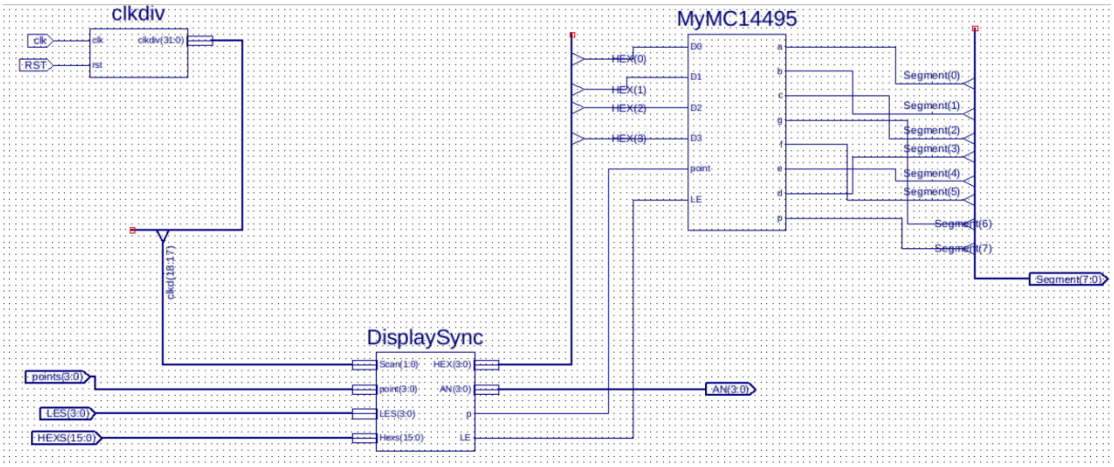
```
//      Verilog      test      fixture      created      from      schematic
/home/ise/Mux4to1b4_sch/Mux4to14b.sch - Mon Oct 31 08:10:59 2022
`timescale 1ns / 1ps
module Mux4to14b_Mux4to14b_sch_tb();
   reg [1:0] S;
   reg [3:0] I0;
   reg [3:0] I1;
   reg [3:0] I2;
   reg [3:0] I3;
   wire [3:0] o;
```

```
   Mux4to14b UUT (
        .S(S),
        .I0(I0),
        .I1(I1),
        .I2(I2),
        .I3(I3),
        .o(o)
   );
integer i;
       initial begin
        S = 0;
        I0 = 1;
        I1 = 2;
        I2 = 4;
        I3 = 8;
        for(S=0;S<=3;S=S+1)
            begin
                #50;
            end
        end
endmodule
```

实验二：记分板设计



DisplaySync 自主作图如上



DispNum 自主作图如上

Top.v Verilog 代码如下：

```
module top(input wire clk,
    input wire [7:0] SW,
    input wire [3:0] btn,
    output wire [3:0] AN,
    output wire [7:0] SEGMENT,
    output wire BTNX4
);
    wire [15:0] num;
    CreateNumber c0(btn,num);
    DispNum d0(clk, num, SW[7:4], SW[3:0], 1'b0, AN, SEGMENT);
     assign BTNX4 = 1'b0; //Enable button inputs
endmodule
```

CreateNumber.v，Verilog 代码如下：

```
module CreateNumber(
        input wire [3:0] btn,
        output reg [15:0] num
    );
wire [3:0] A,B,C,D;
initial num <= 16'b1010_1011_1100_1101;
assign A = num[3:0] + 4'd1;
assign B = num[7:4] + 4'd1;
assign C = num[11:8] + 4'd1;
assign D = num[15:12] + 4'd1;
always @(posedge btn[0]) num[3:0] <= A;
always @(posedge btn[1]) num[7:4] <= B;
always @(posedge btn[2]) num[11:8] <= C;
always @(posedge btn[3]) num[15:12] <= D;
endmodule
```

K7.ucf 如下

```
NET "btn[0]" LOC = W14 | IOSTANDARD = LVCMOS18;
NET "btn[0]" CLOCK_DEDICATED_ROUTE = FALSE;
NET "btn[1]" LOC = V14 | IOSTANDARD = LVCMOS18;
NET "btn[1]" CLOCK_DEDICATED_ROUTE = FALSE;
NET "btn[2]" LOC = V19 | IOSTANDARD = LVCMOS18;
NET "btn[2]" CLOCK_DEDICATED_ROUTE = FALSE;
NET "btn[3]" LOC = V18 | IOSTANDARD = LVCMOS18;
NET "btn[3]" CLOCK_DEDICATED_ROUTE = FALSE;
NET "BTNX4" LOC = W16 | IOSTANDARD = LVCMOS18;
NET "BTN[0]"LOC = AF13 | IOSTANDARD = LVCMOS15;#SW[14]
NET "BTN[1]"LOC = AF10 | IOSTANDARD = LVCMOS15;#SW[15]
NET "SEGMENT[0]"LOC = AB22 | IOSTANDARD = LVCMOS33;#a
NET "SEGMENT[1]" LOC = AD24 | IOSTANDARD = LVCMOS33;#b
NET "SEGMENT[2]" LOC = AD23 | IOSTANDARD = LVCMOS33;#c
NET "SEGMENT[3]" LOC = Y21 | IOSTANDARD = LVCMOS33;#d
NET "SEGMENT[4]" LOC = W20 | IOSTANDARD = LVCMOS33;#e
NET "SEGMENT[5]" LOC = AC24 | IOSTANDARD = LVCMOS33;#f
NET "SEGMENT[6]" LOC = AC23 | IOSTANDARD = LVCMOS33;#g
NET "SEGMENT[7]" LOC = AA22 | IOSTANDARD = LVCMOS33;#point
NET "AN[0]" LOC = AD21 | IOSTANDARD = LVCMOS33;
NET "AN[1]" LOC = AC21 | IOSTANDARD = LVCMOS33;
NET "AN[2]" LOC = AB21 | IOSTANDARD = LVCMOS33;
NET "AN[3]" LOC = AC22 | IOSTANDARD = LVCMOS33;
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 "clk" LOC = AC18 | IOSTANDARD = LVCMOS18;
```

注：该实验已经在 2022.10.31 晚由助教验收通过

三、实验结果与分析

实验结果和预测结果一样，实验操作正确无误，成功地实现了对灯的控制。

四、讨论、心得 （选填）

实验过程中我体会到要认真对待每一次操作，正如我多次因为不小心而操作不熟练使得实验进程受阻。