浙江大学

本科实验报告

课程名称：计算机逻辑设计基础

姓　　名：仇国智

学　　院：竺可桢学院

专　　业：计算机科学与技术

学　　号：3220102181

指导教师：董亚波

2023 年 10 月 28 日

一、 实验目的和要求

1　实验目的

1.1　掌握数据选择器的工作原理和逻辑功能

1.2　掌握数据选择器的使用方法

1.3　掌握 4 位数码管扫描显示方法

1.4　4 位数码管显示应用—记分板设计台

2　实验任务

2.1　任务 1：数据选择器设计

2.2　任务 2：记分板设计

二、 实验内容和原理

1　实验内容

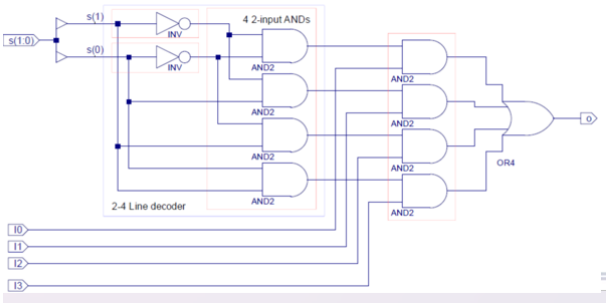
1.1　数据选择器设计

1.2　记分板设计：子任务 1：实现 4 位 7 段数码管动态扫描显示；子任务 2：实现计分板功能。用 BTNX4Y3～BTNX4Y0 这 4 个按钮从左向右分别控制 4 位数码管，每个按钮按一次，对应数码管加 1。

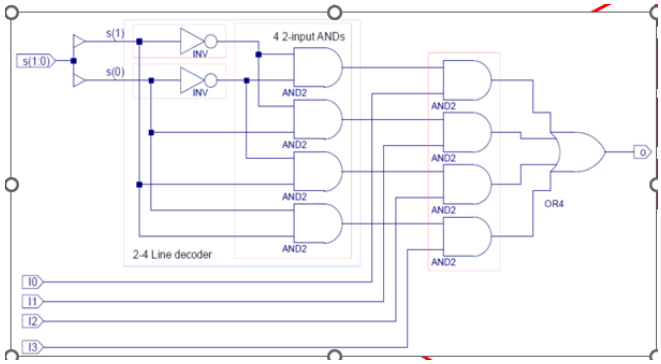
2　实验原理

2.1　选一多路选择器：MUX4to1：根据事件简化真值表。输出是控制信号全部最小项与或结构。

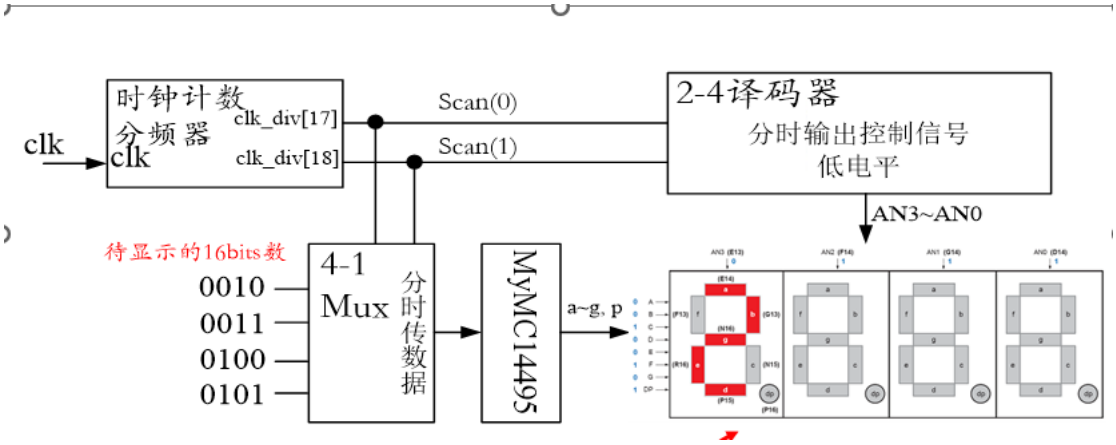
信息输入	控制端	选择输出	
I0 I1 I2 I3	S1 S0	o	输出项
I0 I1 I2 I3	0 0	I0	$\overline{S1}\,\overline{S0}$ I0
I0 I1 I2 I3	0 1	I1	$\overline{S1}S0$ I1
I0 I1 I2 I3	1 0	I2	$S1\overline{S0}$ I2
I0 I1 I2 I3	1 1	I3	S1S0 I3



2.2　控制结构不变，每路输入向量化



2.3 动态扫描显示方案：扫描信号来自时钟计数分频器：时序转化为组合电路。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 Scan[1:0]，其数据为从 0、1、2、3、0、……，输入 2-4 译码器产生数码管位选信号，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）计数器的分频系数要适当，几 ms 切换一次。



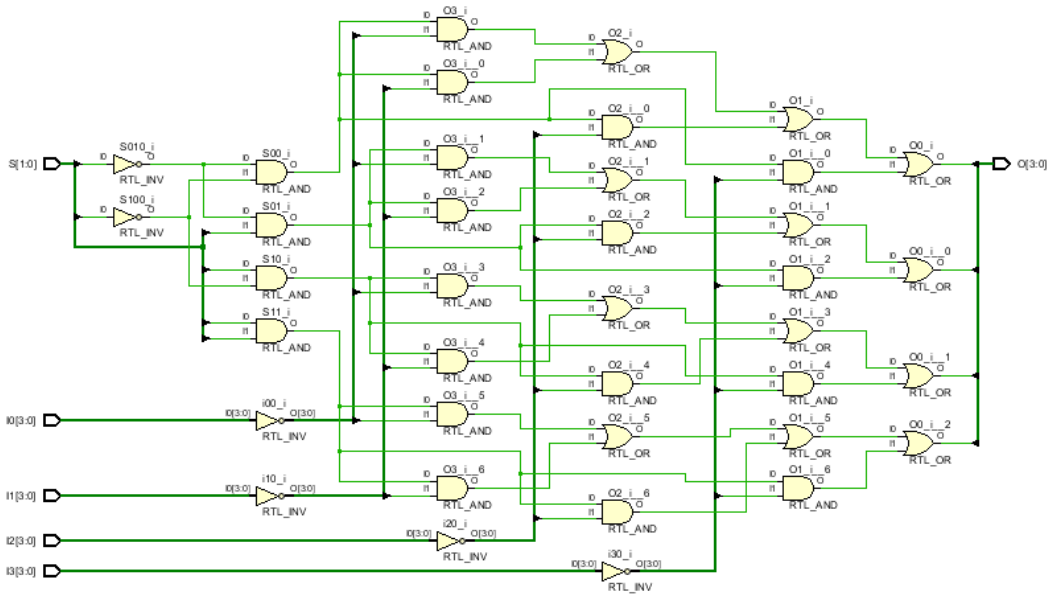
三、 实验过程和数据记录

1 数据选择器设计

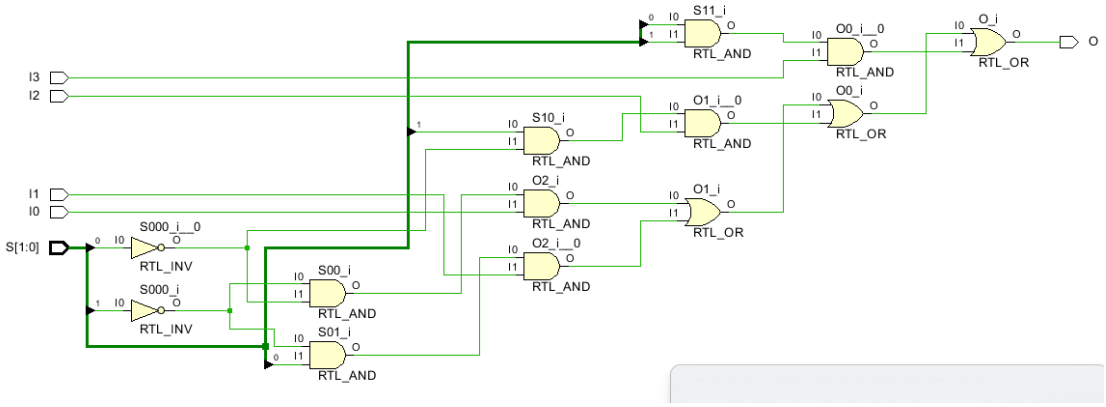
1.1 原理图方式设计 1 位和 4 位 4 选 1 数据选择器

```
`timescale 1ns / 1ps
module Mux4to14b (
    S,
    I0,
    I1,
    I2,
    I3,
    O
);
  input wire [1:0] S;
  input wire [3:0] I0, I1, I2, I3;
  output wire [3:0] O;
  wire S00, S01, S10, S11;
  wire [3:0] i0, i1, i2, i3;
```

```
  assign i0[3:0] = ~I0[3:0], i1[3:0] = ~I1[3:0], i2[3:0] = ~I2[3:0],
i3[3:0] = ~I3[3:0];
  assign S00 = !S[1] & !S[0], S01 = !S[1] & S[0], S10 = S[1] & !S[0],
S11 = S[0] & S[1];
  assign O[0] = S00 & i0[0] | S00 & i1[0] | S00 & i2[0] | S00 & i3[0];
  assign O[1] = S01 & i0[1] | S01 & i1[1] | S01 & i2[1] | S01 & i3[1];
  assign O[2] = S10 & i0[2] | S10 & i1[2] | S10 & i2[2] | S10 & i3[2];
  assign O[3] = S11 & i0[3] | S11 & i1[3] | S11 & i2[3] | S11 & i3[3];
endmodule
```



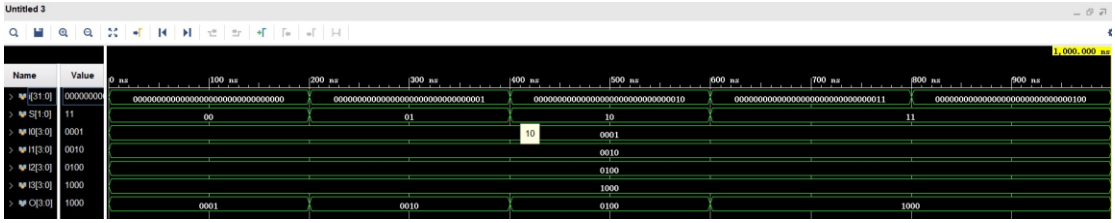

```
module Mux4to1 (
    S,
    I0,
    I1,
    I2,
    I3,
    O
);
  input wire [1:0] S;
  input wire I0, I1, I2, I3;
  output wire O;
  wire S00, S01, S10, S11;
  assign S00 = !S[1] & !S[0], S01 = !S[1] & S[0], S10 = S[1] & !S[0],
S11 = S[0] & S[1];
  assign O = S00 & I0 | S01 & I1 | S10 & I2 | S11 & I3;
endmodule
```



1.2 建立仿真波形文件，初始化 I0、I1、I2、I3 和 S[1:0]，进行仿真 Mux4to14b：

```
`timescale 1ns / 1ps
module test1 ();
  integer i;
  reg [1:0] S;
  reg [3:0] I0, I1, I2, I3;
  wire [3:0] O;
  Mux4to14b test1 (
      S,
      I0,
      I1,
      I2,
      I3,
      O
  );
  initial begin
    I0 = 4'b0001;
    I1 = 4'b0010;
    I2 = 4'b0100;
    I3 = 4'b1000;
    for (i = 0; i < 4; i = i + 1) begin
      S = i;
      #200;
    end
  end
```

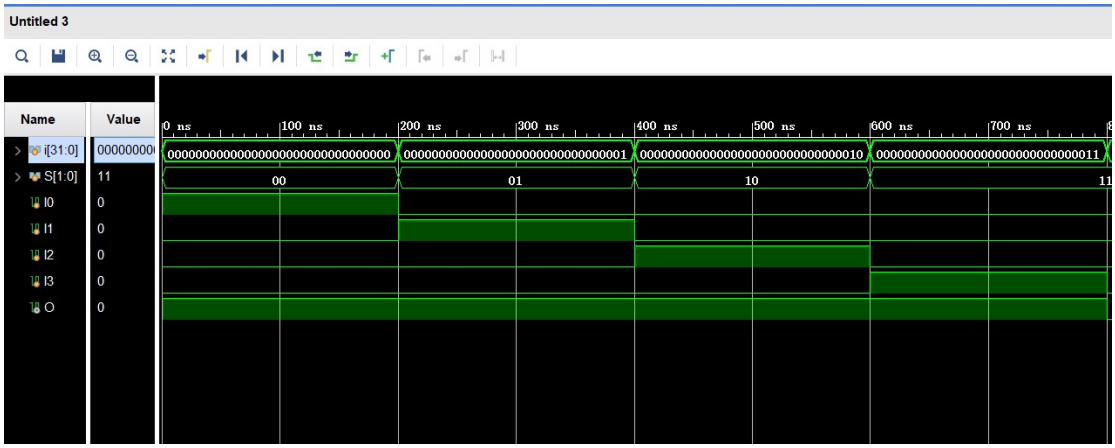
```
endmodule
```



Mux4to1：

```
`timescale 1ns / 1ps
module test2 ();
  integer i;
  reg [1:0] S;
  reg I0, I1, I2, I3;
  wire  O;
  Mux4to1 test2 (
      S,
      I0,
      I1,
      I2,
      I3,
      O
  );
  initial begin
    I0 = 1'b1;
    I1 = 1'b0;
    I2 = 1'b0;
    I3 = 1'b0;
    for (i = 0; i < 4; i = i + 1) begin
      S = i;
      #200;
      {I3,I2,I1,I0}=2*{I3,I2,I1,I0};

    end
  end
endmodule
```



1.3

2　记分板设计

2.1 用原理图形式设计 DisplaySync 模块

输入：

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

输出：

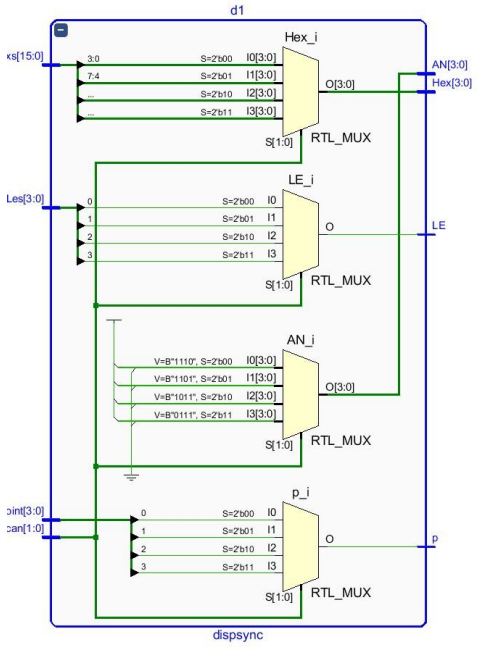
HEX(3:0)：当前要显示的 4 位二进制数

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

```
module dispsync (
    input [15:0] Hexs,
    input [1:0] Scan,
    input [3:0] Point,
    input [3:0] Les,
    output reg [3:0] Hex,
    output reg p,
    output reg LE,
    output reg [3:0] AN
);
  always @* begin
    case (Scan)
      2'b00: begin
        Hex <= Hexs[3:0];
        AN  <= 4'b1110;
        p   <= Point[0];
        LE  <= Les[0];
```

```
      end
      2'b01: begin
        Hex <= Hexs[7:4];
        AN  <= 4'b1101;
        p   <= Point[1];
        LE  <= Les[1];
      end
      2'b10: begin
        Hex <= Hexs[11:8];
        AN  <= 4'b1011;
        p   <= Point[2];
        LE  <= Les[2];
      end
      2'b11: begin
        Hex <= Hexs[15:12];
        AN  <= 4'b0111;
        p   <= Point[3];
        LE  <= Les[3];
      end
    endcase
  end
endmodule
```



2.2 设计 32 位时钟计数分频器

输入：

clk：实验板主时钟

rst：复位信号

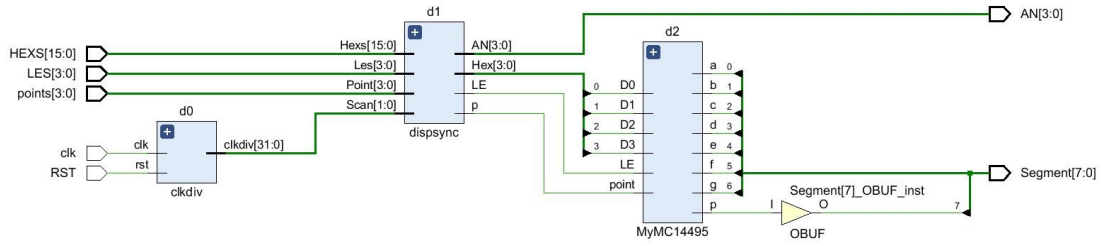
输出：

clkdiv(31:0)：分频时钟输出

```
module clkdiv (
    input clk,
    input rst,
    output reg [31:0] clkdiv
);
  always @(posedge clk or posedge rst) begin
    if (rst) clkdiv <= 0;
    else clkdiv <= clkdiv + 1'b1;
  end
endmodule
```

2.3 设计实现 4 位 7 段数码管动态扫描显示 DispNum 模块

```
`timescale 1ns / 1ps
module DispNum(
clk,
HEXS,
LES,
points,
RST,
AN,
Segment
    );
    input wire clk,RST;
    input wire[3:0]LES,points;
    output wire[3:0]AN;
    input wire[15:0] HEXS;
    output wire[7:0] Segment;
    wire[31:0]clkd;
    wire [3:0]HEX;
    wire point,LE;
    clkdiv d0(clk,RST,clkd);
    dispsync d1(HEXS,clkd[18:17],points,LES,HEX,point,LE,AN);
    MyMC14495
d2(HEX[0],HEX[1],HEX[2],HEX[3],point,LE,Segment[0],Segment[1],Segment[2
],Segment[3],Segment[4],Segment[5],Segment[6],Segment[7]);
endmodule
```



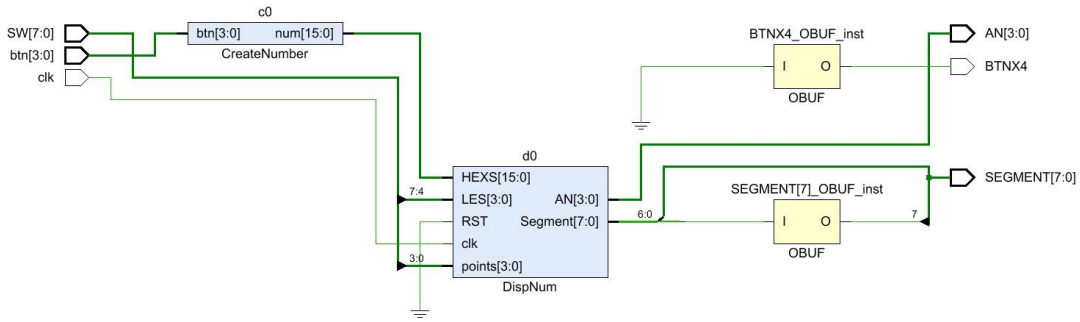
2.4 设计 CreateNumber 按键数据输入模块。使用行为描述设计：四个按键，各按一下，4 个 4 位 2 进制数分别加 1。

```
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
```

2.5 新建源文件 top，在右键菜单里设为“Top Module”

```
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;
 wire gnd;
```

```
  CreateNumber c0 (
      btn,
      num
  );
assign BTNX4=1'b0;
  DispNum d0 (
      clk,
      num,
      SW[7:4],
      SW[3:0],
      1'b0,
      AN,
      SEGMENT
  );
endmodule
```



2.6 编写引脚约束文件

```
set_property PACKAGE_PIN W14 [get_ports {btn[0]}]
set_property PACKAGE_PIN V14 [get_ports {btn[1]}]
set_property PACKAGE_PIN V19 [get_ports {btn[2]}]
set_property PACKAGE_PIN V18 [get_ports {btn[3]}]
set_property PACKAGE_PIN W16 [get_ports BTNX4]
set_property PACKAGE_PIN AC18 [get_ports clk]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SEGMENT[7]}]
```

```
set_property IOSTANDARD LVCMOS33 [get_ports {SEGMENT[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SEGMENT[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SEGMENT[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SEGMENT[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SEGMENT[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SEGMENT[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SEGMENT[0]}]
set_property IOSTANDARD LVCMOS15 [get_ports {SW[7]}]
set_property IOSTANDARD LVCMOS15 [get_ports {SW[6]}]
set_property IOSTANDARD LVCMOS15 [get_ports {SW[5]}]
set_property IOSTANDARD LVCMOS15 [get_ports {SW[4]}]
set_property IOSTANDARD LVCMOS15 [get_ports {SW[3]}]
set_property IOSTANDARD LVCMOS15 [get_ports {SW[2]}]
set_property IOSTANDARD LVCMOS15 [get_ports {SW[1]}]
set_property IOSTANDARD LVCMOS15 [get_ports {SW[0]}]
set_property PACKAGE_PIN AA10 [get_ports {SW[0]}]
set_property PACKAGE_PIN AB10 [get_ports {SW[1]}]
set_property PACKAGE_PIN AA13 [get_ports {SW[2]}]
set_property PACKAGE_PIN AA12 [get_ports {SW[3]}]
set_property PACKAGE_PIN Y13 [get_ports {SW[4]}]
set_property PACKAGE_PIN Y12 [get_ports {SW[5]}]
set_property PACKAGE_PIN AD11 [get_ports {SW[6]}]
set_property PACKAGE_PIN AD10 [get_ports {SW[7]}]
set_property PACKAGE_PIN AB22 [get_ports {SEGMENT[0]}]
set_property PACKAGE_PIN AD24 [get_ports {SEGMENT[1]}]
set_property PACKAGE_PIN AD23 [get_ports {SEGMENT[2]}]
set_property PACKAGE_PIN Y21 [get_ports {SEGMENT[3]}]
set_property PACKAGE_PIN W20 [get_ports {SEGMENT[4]}]
set_property PACKAGE_PIN AC24 [get_ports {SEGMENT[5]}]
set_property PACKAGE_PIN AC23 [get_ports {SEGMENT[6]}]
set_property PACKAGE_PIN AA22 [get_ports {SEGMENT[7]}]
set_property PACKAGE_PIN AD21 [get_ports {AN[0]}]
set_property PACKAGE_PIN AC21 [get_ports {AN[1]}]
set_property PACKAGE_PIN AB21 [get_ports {AN[2]}]
set_property PACKAGE_PIN AC22 [get_ports {AN[3]}]
set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets {btn[0]}]
set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets {btn[1]}]
set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets {btn[2]}]
set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets {btn[3]}]

set_property IOSTANDARD LVCMOS18 [get_ports {btn[3]}]
set_property IOSTANDARD LVCMOS18 [get_ports {btn[2]}]
set_property IOSTANDARD LVCMOS18 [get_ports {btn[1]}]
set_property IOSTANDARD LVCMOS18 [get_ports {btn[0]}]
```

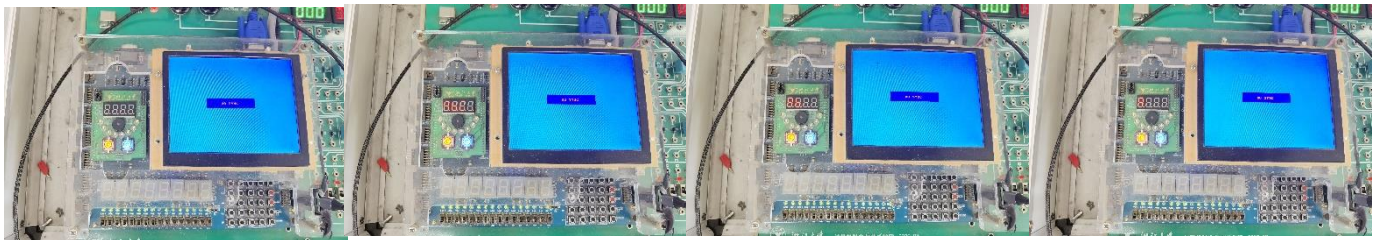
```
set_property IOSTANDARD LVCMOS18 [get_ports BTNX4]
set_property IOSTANDARD LVCMOS18 [get_ports clk]
```

I/O Ports

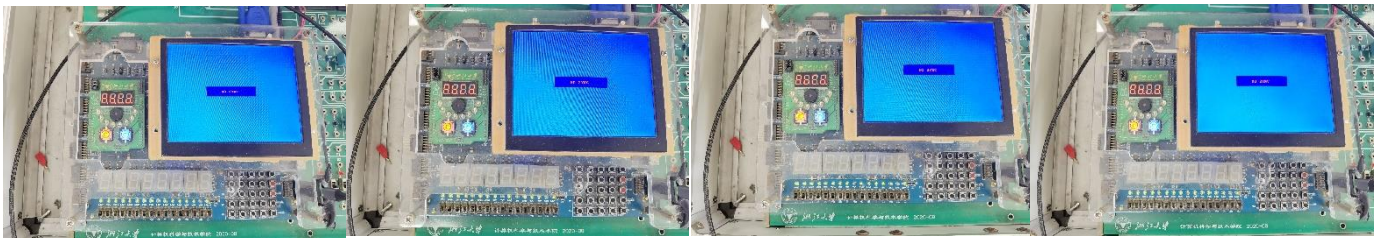
Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination	IN_TERM
All ports (26)													
AN (4)	OUT			☑	12	LVCMOS33*	3.300		12	SLOW	NONE	FP_VTT_50	
AN[3]	OUT		AC22	☑	12	LVCMOS33*	3.300		12	SLOW	NONE	FP_VTT_50	
AN[2]	OUT		AB21	☑	12	LVCMOS33*	3.300		12	SLOW	NONE	FP_VTT_50	
AN[1]	OUT		AC21	☑	12	LVCMOS33*	3.300		12	SLOW	NONE	FP_VTT_50	
AN[0]	OUT		AD21	☑	12	LVCMOS33*	3.300		12	SLOW	NONE	FP_VTT_50	
btn (4)	IN			☑	32	LVCMOS18	1.800				NONE	NONE	
btn[3]	IN		V18	☑	32	LVCMOS18	1.800				NONE	NONE	
btn[2]	IN		V19	☑	32	LVCMOS18	1.800				NONE	NONE	
btn[1]	IN		V14	☑	32	LVCMOS18	1.800				NONE	NONE	
btn[0]	IN		W14	☑	32	LVCMOS18	1.800				NONE	NONE	
SEGMENT (8)	OUT			☑	12	LVCMOS33*	3.300		12	SLOW	NONE	FP_VTT_50	
SEGMEN...	OUT		AA22	☑	12	LVCMOS33*	3.300		12	SLOW	NONE	FP_VTT_50	
SEGMEN...	OUT		AC23	☑	12	LVCMOS33*	3.300		12	SLOW	NONE	FP_VTT_50	
SEGMEN...	OUT		AC24	☑	12	LVCMOS33*	3.300		12	SLOW	NONE	FP_VTT_50	
SEGMEN...	OUT		W20	☑	12	LVCMOS33*	3.300		12	SLOW	NONE	FP_VTT_50	
SEGMEN...	OUT		Y21	☑	12	LVCMOS33*	3.300		12	SLOW	NONE	FP_VTT_50	
SEGMEN...	OUT		AD23	☑	12	LVCMOS33*	3.300		12	SLOW	NONE	FP_VTT_50	
SEGMEN...	OUT		AD24	☑	12	LVCMOS33*	3.300		12	SLOW	NONE	FP_VTT_50	
SEGMEN...	OUT		AB22	☑	12	LVCMOS33*	3.300		12	SLOW	NONE	FP_VTT_50	
SW (8)	IN			☑	33	LVCMOS15*	1.500				NONE	NONE	
SW[7]	IN		AD10	☑	33	LVCMOS15*	1.500				NONE	NONE	
SW[6]	IN		AD11	☑	33	LVCMOS15*	1.500				NONE	NONE	
SW[5]	IN		Y12	☑	33	LVCMOS15*	1.500				NONE	NONE	
SW[4]	IN		Y13	☑	33	LVCMOS15*	1.500				NONE	NONE	
SW[3]	IN		AA12	☑	33	LVCMOS15*	1.500				NONE	NONE	
SW[2]	IN		AA13	☑	33	LVCMOS15*	1.500				NONE	NONE	
SW[1]	IN		AB10	☑	33	LVCMOS15*	1.500				NONE	NONE	
SW[0]	IN		AA10	☑	33	LVCMOS15*	1.500				NONE	NONE	
Scalar ports (2)													
BTNX4	OUT		W16	☑	32	LVCMOS18	1.800		12	SLOW	NONE	FP_VTT_50	
clk	IN		AC18	☑	32	LVCMOS18	1.800				NONE	NONE	

2.7 生成并下载 bit 文件，进行实际测试

测试小数点控制：

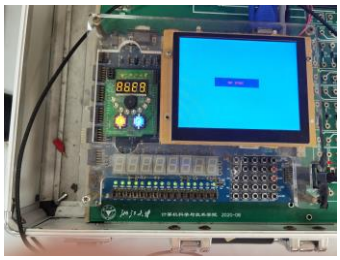
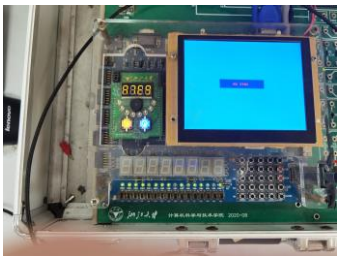


测试数字消隐：



测试按键数字递增：



四、 实验结果分析

波形图同 ppt 相同，输出随 O 随选择信号线 S 在 I0,I1,I2,I3 中依次选择输出；下载验证符合预期，小数点随开关拨动依次亮起，数字随开关拨动依次消隐，每按一次键数字会发生变化。但是唯一的不足是数字每次不是只变化 1。

五、 讨论与心得

按钮按下过程中电压不稳定，会导致数字递增多次，可以使用开关来稳定，或者采用其它方式避免电压波动造成的影响。