

浙江大学

本科实验报告

课程名称：数字逻辑设计

姓名：沈一芑

学院：计算机学院

系：计算机系

专业：计算机科学与技术

学号：3220101827

指导教师：马德

2023 年 11 月 6 日

浙江大学实验报告

课程名称: 数字逻辑设计 实验类型: ISE 电路设计

实验项目名称: 多路选择器设计与应用

学生姓名: 沈一芃 专业: 计算机科学与技术 学号: 322010827

同组学生姓名: 无 指导老师: 马德

实验地点: 东 4 509 实验日期: 2023 年 11 月 6 日

一、实验目的和要求:

- 掌握数据选择器的工作原理和逻辑功能
- 掌握数据选择器的使用方法
- 掌握 4 位数码管扫描显示方法
- 4 位数码管显示应用—记分板设计

二、实验内容和原理

1. 实验内容:

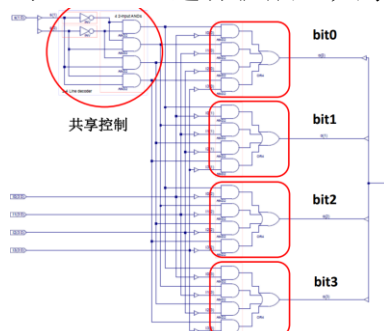
- 数据选择器设计
- 记分板设计

2. 实验原理:

- MUX4to1 多路选择器的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 $\overline{S1S0}$
I0 I1 I2 I3	0 1	I1 $\overline{S1S0}$
I0 I1 I2 I3	1 0	I2 $\overline{S1S0}$
I0 I1 I2 I3	1 1	I3 $\overline{S1S0}$

- 对 MUX4to1 进行扩展, 共享控制, 扩展 bit 位, 得到 MUX4to1b4



三、 主要仪器设备

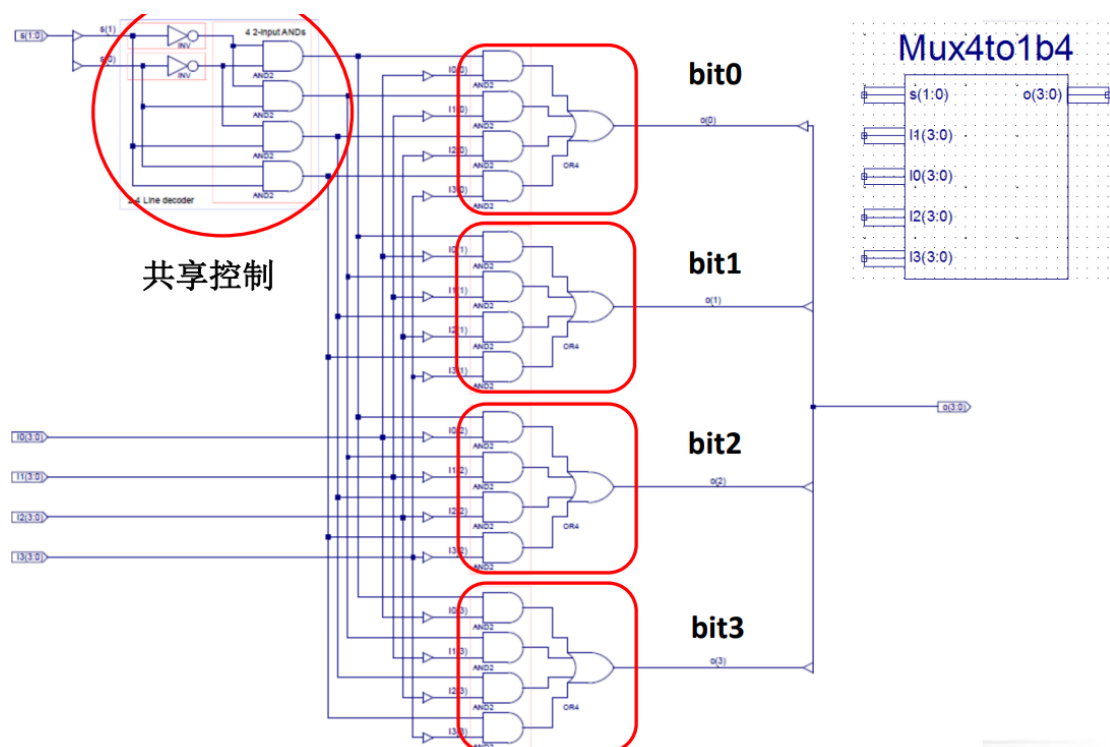
实验设备：

- 装有 Xilinx ISE 14.7 的计算机 1 台
- SWORD 开发板 1 套

四、 操作方法与实验步骤

1. 数据选择器设计

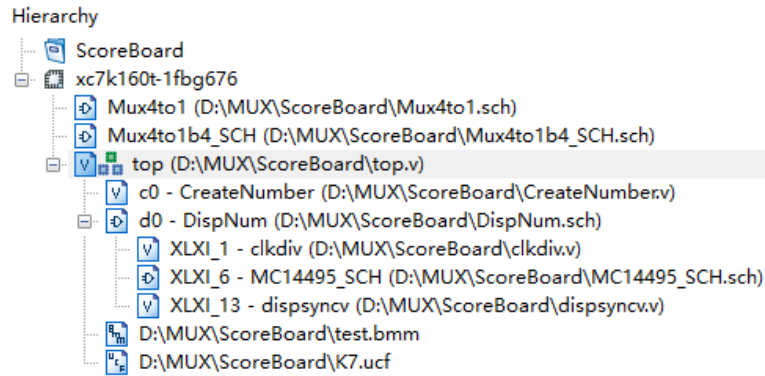
- 建立工程 MUX4to1b4
- 建立名为 MUX4t1b4 的 schematic 源文件
- 原理图方式进行设计



- 建立测试波形文件 MUX4t1b4_sim.tbw 并进行仿真激励输入。在 simulation 视图中 选择 simulate behavioral model, 并验证正确性:
- 通过 Create Schematic Symbol 生成 MUX4t1b4 模块的逻辑符号图文件, 后缀.sym。

2. 计分板设计

- 新建工程 ScoreBoard, Top level source type 选用 HDL。
- 设计动态扫描同步输出模块。工程结构如下:



- 设计顶层模块

```
21 module top(
22     input wire clk,
23     input wire [7:0] SW,
24     input wire [3:0] btn,
25     output wire [3:0] AN,
26     output wire [7:0] SEGMENT,
27     output wire BTNX4
28 );
29
30     wire [15:0] num;
31     CreateNumber c0(btn, num);
32     DispNum d0(clk, num, SW[7:4], SW[3:0], 1'b0, AN, SEGMENT);
33     assign BTNX4 = 1'b0; //enable button inputs
34
35 endmodule
```

- 设计 CreateNumber 模块

```
21 module CreateNumber(
22     input wire [3:0] btn,
23     output reg [15:0] num
24 );
25     wire [3:0] A, B, C, D;
26     initial num <= 16'b0001_0010_0011_0100; //display 'abcd'
27
28     assign A = num [ 3: 0] + 4'd1;
29     assign B = num [ 7: 4] + 4'd1;
30     assign C = num [11: 8] + 4'd1;
31     assign D = num [15:12] + 4'd1;
32
33     always@(posedge btn[0]) num[ 3: 0] <= A;
34     always@(posedge btn[1]) num[ 7: 4] <= B;
35     always@(posedge btn[2]) num[11: 8] <= C;
36     always@(posedge btn[3]) num[15:12] <= D;
37
38 endmodule
```

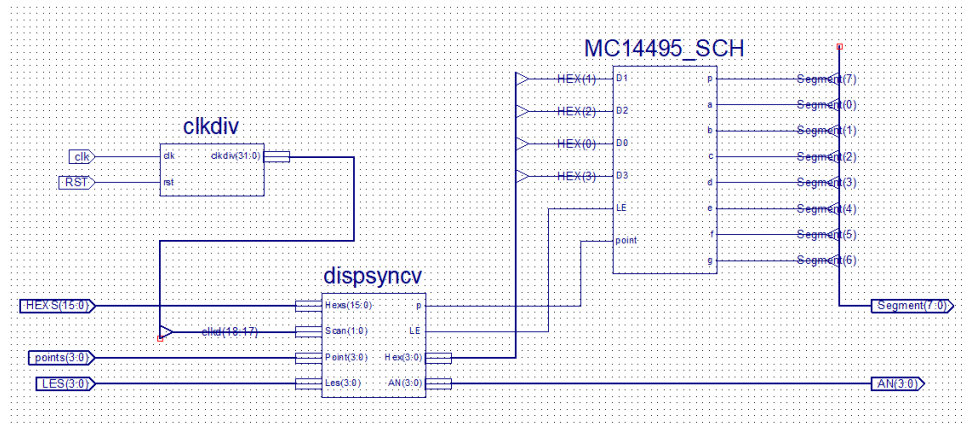
- 设计 DispSync 模块

```

21 module dispyncv(
22     input [15:0] Hexas,
23     input [1:0] Scan,
24     input [3:0] Point,
25     input [3:0] Les,
26     output reg[3:0] Hex,
27     output reg p,LE,
28     output reg[3:0] AN
29 );
30
31     always @* begin
32         case(Scan)
33             2'b00 : begin Hex <= Hexas[ 3: 0];   AN <= 4'b1110; p <= Point[0]; LE <= Les[0]; end
34             2'b01 : begin Hex <= Hexas[ 7: 4];   AN <= 4'b1101; p <= Point[1]; LE <= Les[1]; end
35             2'b10 : begin Hex <= Hexas[11: 8];   AN <= 4'b1011; p <= Point[2]; LE <= Les[2]; end
36             2'b11 : begin Hex <= Hexas[15:12];   AN <= 4'b0111; p <= Point[3]; LE <= Les[3]; end
37         endcase
38     end
39
40 endmodule

```

- 设计时钟计数分频器，并建立 DispNum 模块



- 建立约束文件，生成执行文件

```

1  NET "btn[0]" LOC = W14 | IOSTANDARD = LVCMOS18;
2  NET "btn[0]" CLOCK_DEDICATED_ROUTE = FALSE;
3  NET "btn[1]" LOC = V14 | IOSTANDARD = LVCMOS18;
4  NET "btn[1]" CLOCK_DEDICATED_ROUTE = FALSE;
5  NET "btn[2]" LOC = V19 | IOSTANDARD = LVCMOS18;
6  NET "btn[2]" CLOCK_DEDICATED_ROUTE = FALSE;
7  NET "btn[3]" LOC = V18 | IOSTANDARD = LVCMOS18;
8  NET "btn[3]" CLOCK_DEDICATED_ROUTE = FALSE;
9  NET "BTN4" LOC = W16 | IOSTANDARD = LVCMOS18;
10
11 NET "clk" LOC = AC18 | IOSTANDARD = LVCMOS18;
12 NET "SW[0]" LOC = AA10 | IOSTANDARD = LVCMOS15;
13 NET "SW[1]" LOC = AB10 | IOSTANDARD = LVCMOS15;
14 NET "SW[2]" LOC = AA13 | IOSTANDARD = LVCMOS15;
15 NET "SW[3]" LOC = AA12 | IOSTANDARD = LVCMOS15;
16 NET "SW[4]" LOC = Y13 | IOSTANDARD = LVCMOS15;
17 NET "SW[5]" LOC = Y12 | IOSTANDARD = LVCMOS15;
18 NET "SW[6]" LOC = AD11 | IOSTANDARD = LVCMOS15;
19 NET "SW[7]" LOC = AD10 | IOSTANDARD = LVCMOS15;
20
21 NET "SEGMENT[0]" LOC = AB22 | IOSTANDARD = LVCMOS33;#a
22 NET "SEGMENT[1]" LOC = AD24 | IOSTANDARD = LVCMOS33;#b
23 NET "SEGMENT[2]" LOC = AD23 | IOSTANDARD = LVCMOS33;#c
24 NET "SEGMENT[3]" LOC = Y21 | IOSTANDARD = LVCMOS33;#d
25 NET "SEGMENT[4]" LOC = W20 | IOSTANDARD = LVCMOS33;#e
26 NET "SEGMENT[5]" LOC = AC24 | IOSTANDARD = LVCMOS33;#f
27 NET "SEGMENT[6]" LOC = AC23 | IOSTANDARD = LVCMOS33;#g
28 NET "SEGMENT[7]" LOC = AA22 | IOSTANDARD = LVCMOS33;#p
29
30 NET "AN[0]" LOC = AD21 | IOSTANDARD = LVCMOS33;
31 NET "AN[1]" LOC = AC21 | IOSTANDARD = LVCMOS33;
32 NET "AN[2]" LOC = AB21 | IOSTANDARD = LVCMOS33;
33 NET "AN[3]" LOC = AC22 | IOSTANDARD = LVCMOS33;

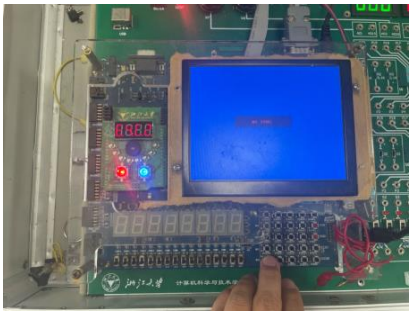
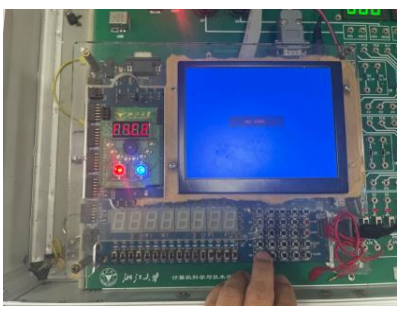
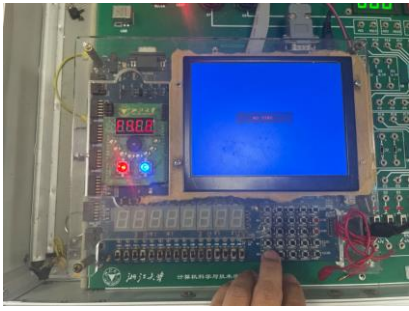
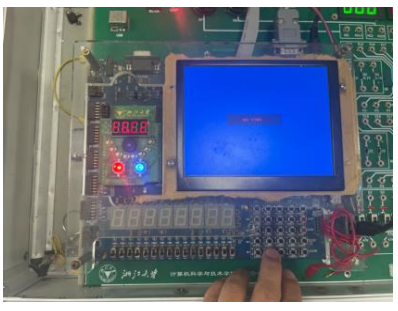

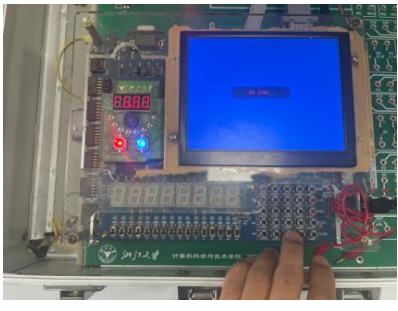
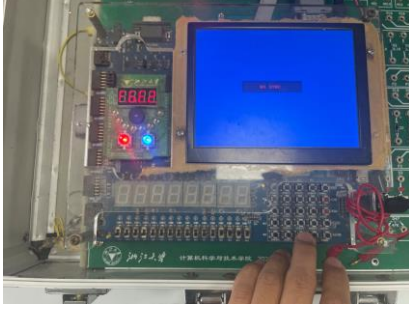

```

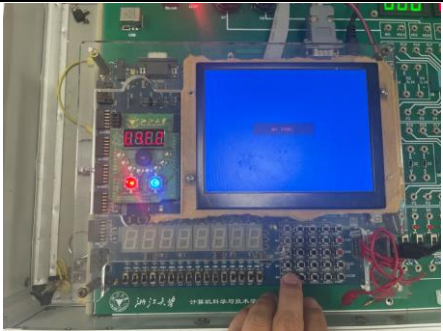
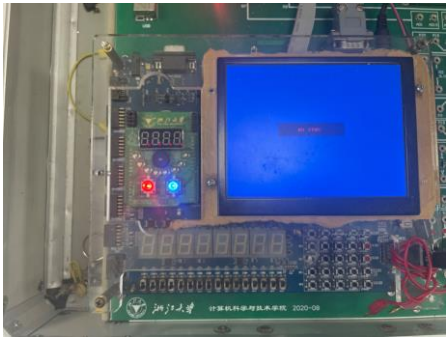
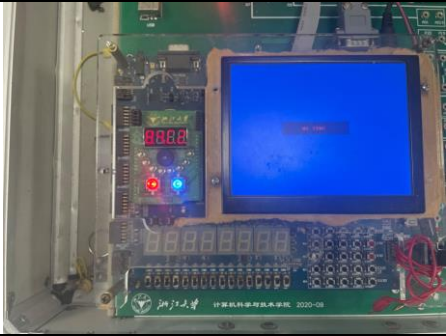
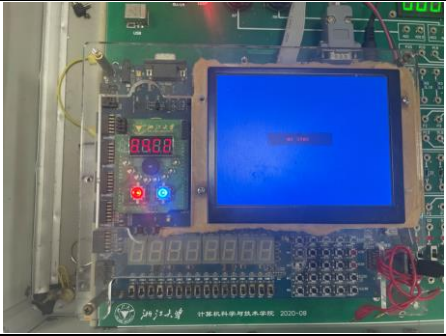
- 下载 bit 文件至 sword 板，根据真值表进行功能验证。

五、 实验数据记录和处理

1. 数据选择器设计

实验截图如下：

按 动 btn0		
按 动 btn1		
按 动 btn2		
按 动 btn3		

数字显示		
数字隐藏		
小数点显示		
小数点隐藏		

六、 实验结果与分析

1. 数据选择器设计

按动 btn0	E4C2 -> F4C2
按动 btn0	F4C2 -> F6C2
按动 btn0	F6C2 -> F6E2
按动 btn0	F6E2 -> F6E4
数字消失	实现
小数点消失	实现

功能正确。设计正确。

七、 讨论、心得

本次实验我掌握了多路选择器的设计，并第一次设计完成一个较为复杂的功能电路：计分板。

设计多路选择器较为简单。吸取上次实验的经验，我在设计好后自己写好仿真激励，进行验证，发现多路选择器设计正确。在此之后才将其保存为一个模块。

设计计分板较为复杂。我首先理清电路关系，手工画出电路总图，明确了各部件之间的关系。随后我对每一个模块从小到大分别进行设计。对于 DispSync 部分，我采取了 .sch 与 .v 两种方式进行设计，相互对照验证功能正确，并将其作为 DispNum 的一部分嵌入模块中。最后，我将 top 选择设定为最顶层设计，在其中调用、连接各个模块，完成计分板电路的设计。按下 btn0-3 的按钮可以在不同七段管上各自加一。

而本次是实验最大的问题是，btn 按钮的按动判定并不灵敏，会有多次判定或未成功触发的情况，导致测试结果较乱。这一问题可以在下一个实验中通过设计消抖模块来弥补。

总得来看，通过了解、设计多路选择器和计分板，我对数字逻辑设计有了更深入的理解，也更加熟悉了 Xilinx ISE 环境及 SWORD 实验平台的使用。这次我

也真正体验到一个较为复杂的功能电路的设计流程。本次实验是一个需要大量思考又十分有意思有收获的实验。希望我能从中汲取更多智慧。

实验日期：2023.11.6