

实验六

课程名称： 数字逻辑电路设计 实验类型： 综合

实验项目名称： 七段数码管显示译码器设计与应用

学生姓名： 段皞一 学号： 3190105359 同组学生姓名： 无

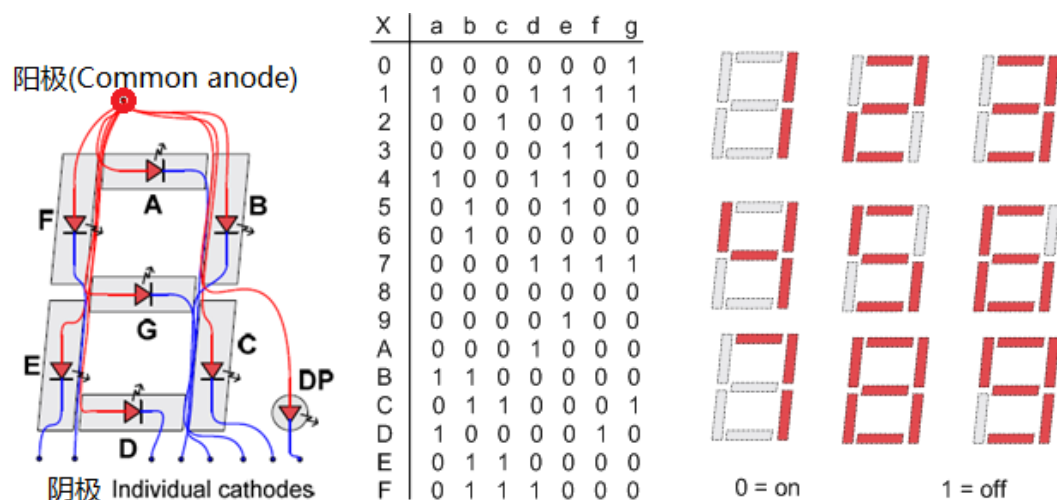
实验地点： 紫金港东四 509 室 实验日期： 2020 年 10 月 27 日

一、操作方法与实验步骤

1.1 实验原理

1.1.1 LED 灯设计

由 7+1 个 LED 构成的数字显示器件；每个 LED 显示数字的一段，另一个为小数点。

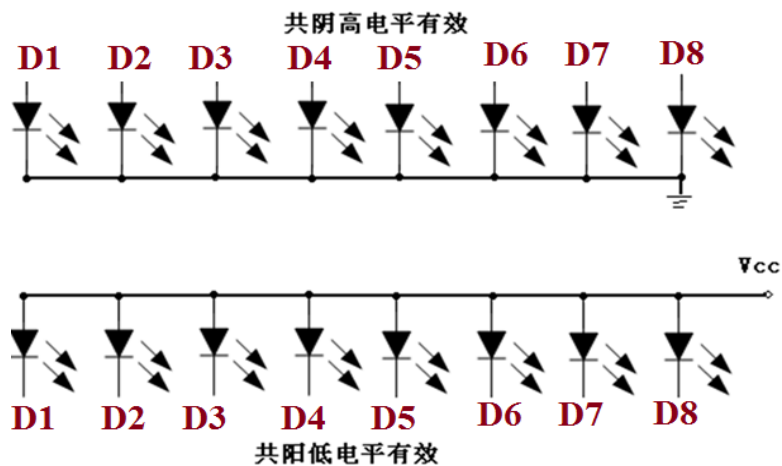


1.1.2 共阴（阳）控制

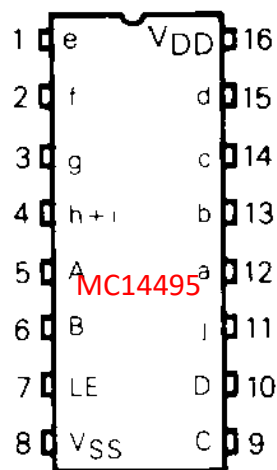
正负逻辑关系说明：

（负逻辑）共阳连接：8 个 LED 正极连在一起，负极低电平时点亮

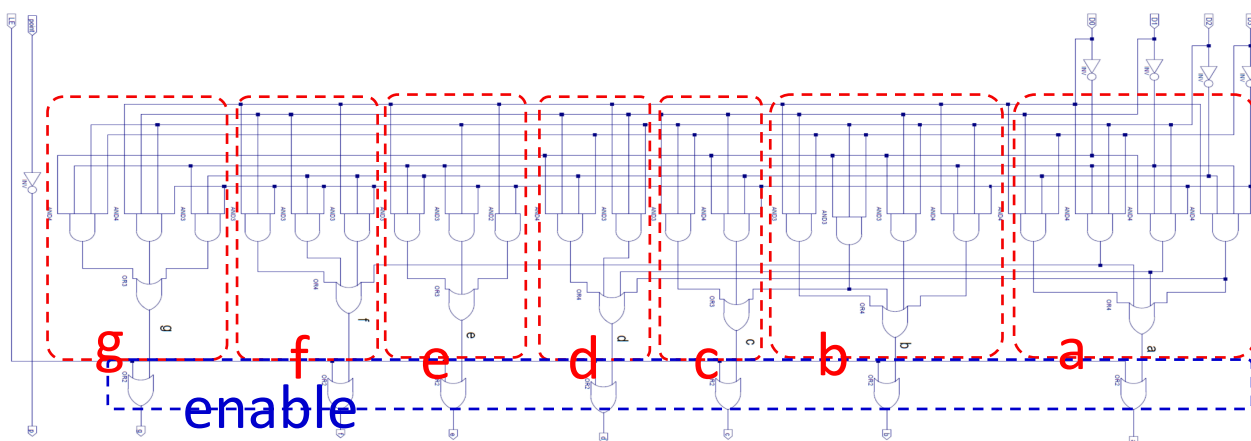
（正逻辑）共阴连接：8 个 LED 负极连在一起，正极高电平时点亮



1.1.2 Hex 7-segment decoder



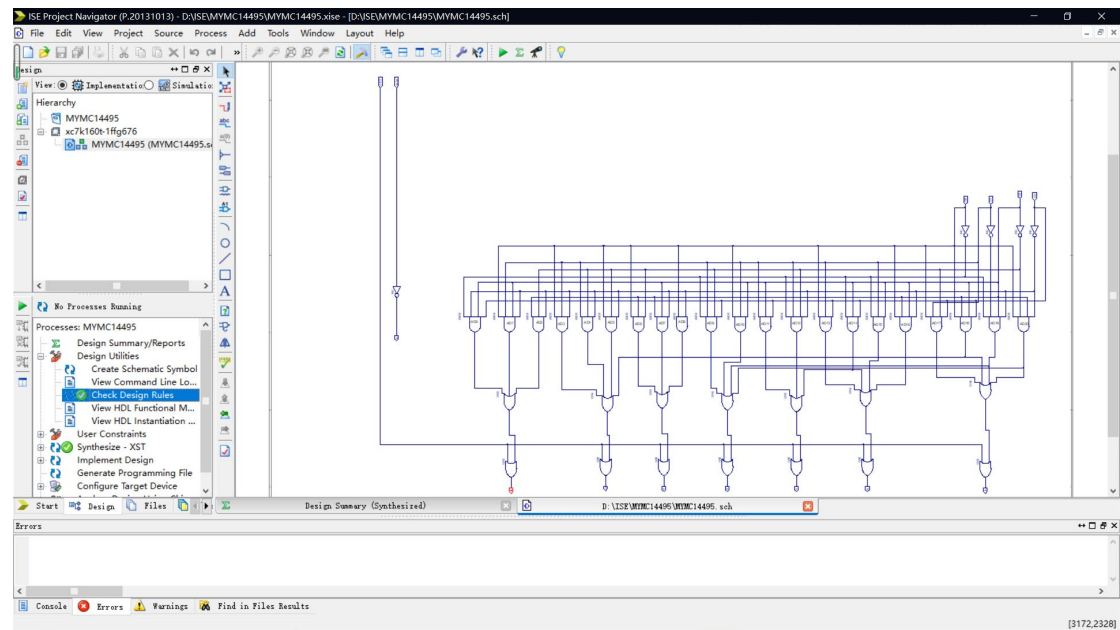
Hex	D ₃ D ₂ D ₁ D ₀	BI/LE	a	b	c	d	e	f	g	p
0	0 0 0 0	1	0	0	0	0	0	0	1	p
1	0 0 0 1	1	1	0	0	1	1	1	1	p
2	0 0 1 0	1	0	0	1	0	0	1	0	p
3	0 0 1 1	1	0	0	0	0	1	1	0	p
4	0 1 0 0	1	1	0	0	1	1	0	0	p
5	0 1 0 1	1	0	1	0	0	1	0	0	p
6	0 1 1 0	1	0	1	0	0	0	0	0	p
7	0 1 1 1	1	0	0	0	1	1	1	1	p
8	1 0 0 0	1	0	0	0	0	0	0	0	P
9	1 0 0 1	1	0	0	0	0	1	0	0	P
A	1 0 1 0	1	0	0	0	1	0	0	0	P
B	1 0 1 1	1	1	1	0	0	0	0	0	P
C	1 1 0 0	1	0	1	1	0	0	0	1	P
D	1 1 0 1	1	1	0	0	0	0	1	0	P
E	1 1 1 0	1	0	1	1	0	0	0	0	P
F	1 1 1 1	1	0	1	1	1	0	0	0	P
X	x x x x	0	1	1	1	1	1	1	1	1



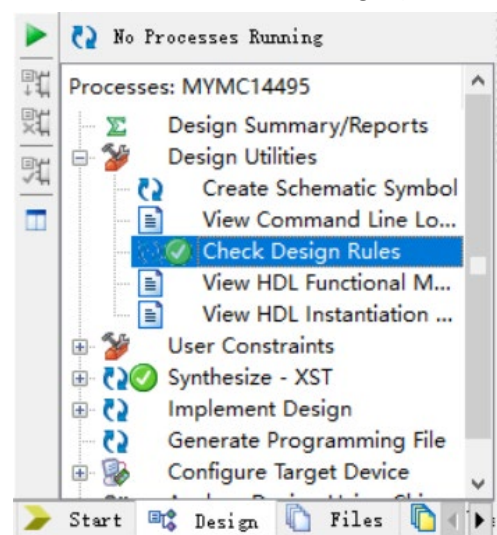
1.2 任务一——原理图设计实现显示译码 MyMC14495 模块

1. 新建名称为“MyMC14495”的工程。
2. 新建名称为“MyMC14495”的 Schematic 文件。

3. 按照原理图进行设计。



4. 在 Processes 窗口中双击 Check Design Rules 无误后，双击 View HDL Functional Model，观看并学习生产成的 Verilog 代码。



```

/////////////////////////////////////////////////////////////////
module MYMC14495(D0,
                D1,
                D2,
                D3,
                LED,
                point,
                a,
                b,
                c,
                d,
                e,
                f,

```

```
g,  
p);
```

```
input D0;  
input D1;  
input D2;  
input D3;  
input LED;  
input point;  
output a;  
output b;  
output c;  
output d;  
output e;  
output f;  
output g;  
output p;
```

```
wire XLXN_47;  
wire XLXN_48;  
wire XLXN_49;  
wire XLXN_50;  
wire XLXN_51;  
wire XLXN_52;  
wire XLXN_53;  
wire XLXN_54;  
wire XLXN_55;  
wire XLXN_56;  
wire XLXN_57;  
wire XLXN_58;  
wire XLXN_61;  
wire XLXN_62;  
wire XLXN_63;  
wire XLXN_64;  
wire XLXN_65;  
wire XLXN_66;  
wire XLXN_67;  
wire XLXN_68;  
wire XLXN_69;  
wire XLXN_74;  
wire XLXN_75;  
wire XLXN_76;  
wire XLXN_77;  
wire XLXN_79;
```

```

wire XLXN_80;
wire XLXN_81;
wire XLXN_84;
wire XLXN_86;
wire XLXN_95;
wire XLXN_98;

AND4 XLXI_7 (.I0(D0),
             .I1(D2),
             .I2(D2),
             .I3(D3),
             .O(XLXN_49));
AND4 XLXI_10 (.I0(XLXN_86),
              .I1(XLXN_98),
              .I2(D2),
              .I3(XLXN_95),
              .O(XLXN_48));
AND4 XLXI_11 (.I0(D0),
              .I1(D1),
              .I2(XLXN_84),
              .I3(D3),
              .O(XLXN_50));
AND4 XLXI_12 (.I0(D0),
              .I1(XLXN_84),
              .I2(D2),
              .I3(XLXN_95),
              .O(XLXN_47));
AND4 XLXI_14 (.I0(XLXN_86),
              .I1(XLXN_95),
              .I2(D2),
              .I3(D3),
              .O(XLXN_68));
AND4 XLXI_22 (.I0(XLXN_86),
              .I1(D1),
              .I2(XLXN_84),
              .I3(XLXN_95),
              .O(XLXN_55));
AND4 XLXI_23 (.I0(XLXN_86),
              .I1(D1),
              .I2(XLXN_84),
              .I3(D3),
              .O(XLXN_57));
AND4 XLXI_24 (.I0(D0),
              .I1(D2),

```

```

        .I2 (D2) ,
        .I3 (XLXN_95) ,
        .O (XLXN_51)) ;
AND4  XLXI_29 (.I0 (D0) ,
        .I1 (D1) ,
        .I2 (D2) ,
        .I3 (XLXN_95) ,
        .O (XLXN_67)) ;
AND3  XLXI_62 (.I0 (XLXN_86) ,
        .I1 (D1) ,
        .I2 (D2) ,
        .O (XLXN_52)) ;
AND3  XLXI_63 (.I0 (XLXN_86) ,
        .I1 (D2) ,
        .I2 (D3) ,
        .O (XLXN_53)) ;
AND3  XLXI_64 (.I0 (D0) ,
        .I1 (D1) ,
        .I2 (D3) ,
        .O (XLXN_54)) ;
AND3  XLXI_65 (.I0 (D1) ,
        .I1 (D2) ,
        .I2 (D3) ,
        .O (XLXN_56)) ;
AND3  XLXI_66 (.I0 (D0) ,
        .I1 (D1) ,
        .I2 (D2) ,
        .O (XLXN_58)) ;
AND2  XLXI_67 (.I0 (D0) ,
        .I1 (XLXN_95) ,
        .O (XLXN_63)) ;
AND3  XLXI_68 (.I0 (D2) ,
        .I1 (D2) ,
        .I2 (XLXN_95) ,
        .O (XLXN_61)) ;
AND3  XLXI_69 (.I0 (D0) ,
        .I1 (D2) ,
        .I2 (XLXN_84) ,
        .O (XLXN_62)) ;
AND3  XLXI_70 (.I0 (D0) ,
        .I1 (D1) ,
        .I2 (XLXN_95) ,
        .O (XLXN_66)) ;
AND3  XLXI_71 (.I0 (D1) ,

```

```

        .I1 (XLXN_84) ,
        .I2 (XLXN_95) ,
        .O (XLXN_65)) ;
AND3  XLXI_72 (.I0 (D0) ,
        .I1 (XLXN_84) ,
        .I2 (XLXN_95) ,
        .O (XLXN_64)) ;
AND3  XLXI_73 (.I0 (D2) ,
        .I1 (XLXN_84) ,
        .I2 (XLXN_95) ,
        .O (XLXN_69)) ;
OR4   XLXI_74 (.I0 (XLXN_50) ,
        .I1 (XLXN_49) ,
        .I2 (XLXN_48) ,
        .I3 (XLXN_47) ,
        .O (XLXN_81)) ;
OR4   XLXI_75 (.I0 (XLXN_54) ,
        .I1 (XLXN_53) ,
        .I2 (XLXN_52) ,
        .I3 (XLXN_51) ,
        .O (XLXN_80)) ;
OR4   XLXI_76 (.I0 (XLXN_57) ,
        .I1 (XLXN_58) ,
        .I2 (XLXN_48) ,
        .I3 (XLXN_47) ,
        .O (XLXN_77)) ;
OR4   XLXI_77 (.I0 (XLXN_66) ,
        .I1 (XLXN_65) ,
        .I2 (XLXN_64) ,
        .I3 (XLXN_49) ,
        .O (XLXN_75)) ;
OR3   XLXI_78 (.I0 (XLXN_56) ,
        .I1 (XLXN_55) ,
        .I2 (XLXN_53) ,
        .O (XLXN_79)) ;
OR3   XLXI_79 (.I0 (XLXN_62) ,
        .I1 (XLXN_61) ,
        .I2 (XLXN_63) ,
        .O (XLXN_76)) ;
OR3   XLXI_80 (.I0 (XLXN_68) ,
        .I1 (XLXN_67) ,
        .I2 (XLXN_69) ,
        .O (XLXN_74)) ;
INV   XLXI_81 (.I (point) ,

```

```

        .O(p));
OR2  XLXI_82 (.IO(LED),
             .I1(XLXN_74),
             .O(g));
OR2  XLXI_83 (.IO(LED),
             .I1(XLXN_75),
             .O(f));
OR2  XLXI_84 (.IO(LED),
             .I1(XLXN_76),
             .O(e));
OR2  XLXI_85 (.IO(LED),
             .I1(XLXN_77),
             .O(d));
OR2  XLXI_86 (.IO(LED),
             .I1(XLXN_79),
             .O(c));
OR2  XLXI_87 (.IO(LED),
             .I1(XLXN_80),
             .O(b));
OR2  XLXI_88 (.IO(LED),
             .I1(XLXN_81),
             .O(a));
INV  XLXI_89 (.I(D0),
             .O(XLXN_86));
INV  XLXI_90 (.I(D1),
             .O(XLXN_98));
INV  XLXI_91 (.I(D2),
             .O(XLXN_84));
INV  XLXI_92 (.I(D3),
             .O(XLXN_95));

endmodule

```

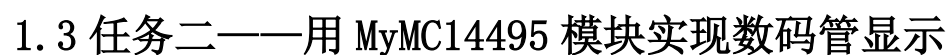
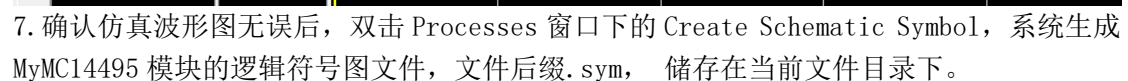
5. 仿真。对 MyMC14495 模块进行仿真，参考激励代码如下：

```

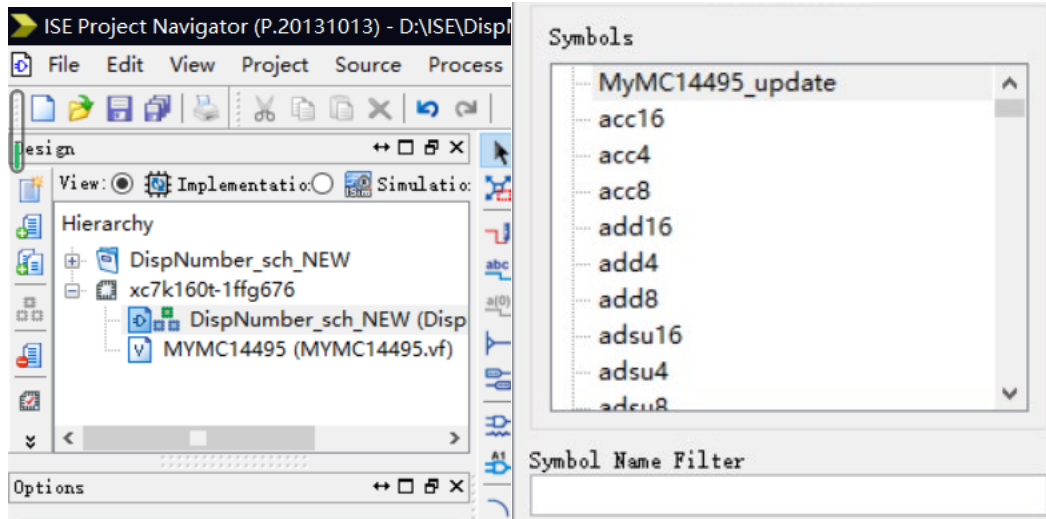
... ..
integer i;
initial begin
    D3 = 0;
    D2 = 0;
    D1 = 0;
    D0 = 0;
    LED = 0;
    point = 0;
    for (i=0; i<=15;i=i+1) begin
        #50;
        {D3,D2,D1,D0}=i;
    end
end

```

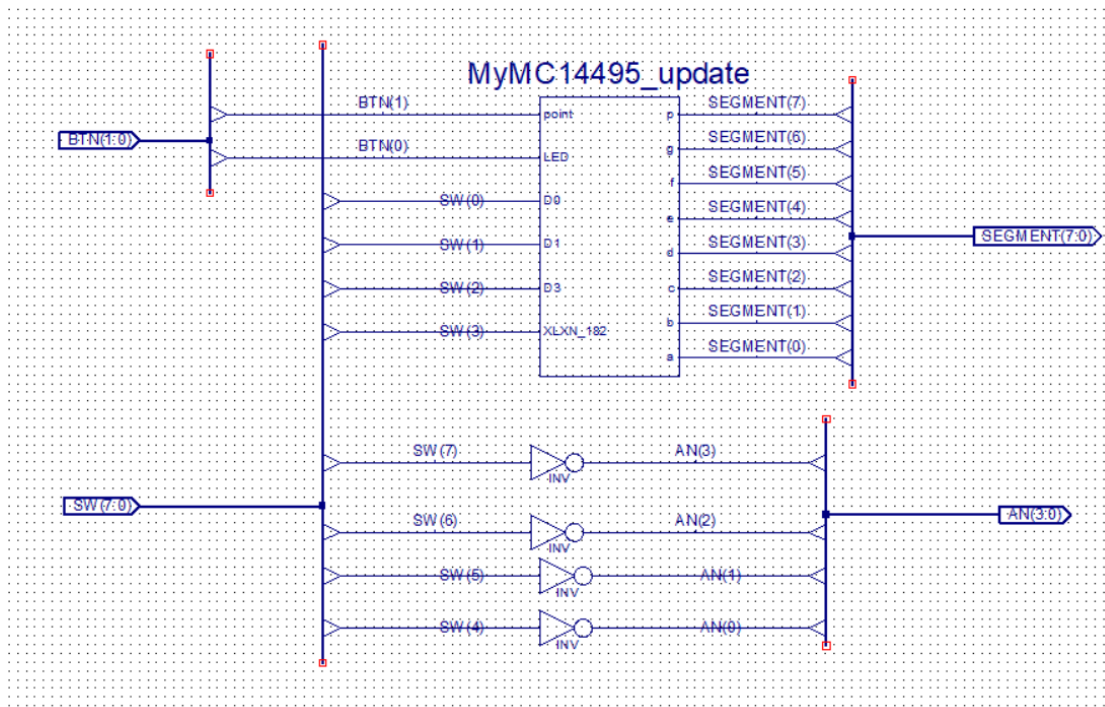

6.生成仿真波形图。



1. 新建工程，文件名为 DispNumber_sch。
2. 新建 Schematic 文件，命名为 DispNumber_sch。
3. 将任务一生成的文件 MyMC14495.sym 和 MyMC14495 复制到任务二工程的根目录下。可以发现，Symbols 框里的第一个文件，就是 MyMC14495。



4. 在文件 DispNumber_sch 中调用逻辑元件 MyMC14495, 完成数码管显示的实验电路连接。



该电路图生成的 fundamental model 如下：

```
`timescale 1ns / 1ps
```

```

module DispNumber_sch(BTN,
                      SW,
                      AN,
                      SEGMENT);

    input [1:0] BTN;
    input [7:0] SW;
    output [3:0] AN;
    output [7:0] SEGMENT;

    MYMC14495 XLXI_1 (.D0(SW[0]),
                    .D1(SW[1]),
                    .D2(SW[2]),
                    .D3(SW[3]),
                    .LED(BTN[0]),
                    .point(BTN[1]),
                    .a(SEGMENT[0]),
                    .b(SEGMENT[1]),
                    .c(SEGMENT[2]),
                    .d(SEGMENT[3]),
                    .e(SEGMENT[4]),
                    .f(SEGMENT[5]),
                    .g(SEGMENT[6]),
                    .p(SEGMENT[7]));

    INV XLXI_2 (.I(SW[7]),
               .O(AN[3]));
    INV XLXI_3 (.I(SW[6]),
               .O(AN[2]));
    INV XLXI_4 (.I(SW[5]),
               .O(AN[1]));
    INV XLXI_5 (.I(SW[4]),
               .O(AN[0]));

Endmodule

```

5.配置电路的引脚。

❑ UCF 引脚定义

■ 输入

- ❑ SW[7:4]=AN[3:0]
- ❑ SW[3:0]=D3D2D1D0
- ❑ SW[14]=LE
- ❑ SW[15]=point

■ 输出

- ❑ a~g, p

引脚配置文件 UCF.ucf 的代码如下：

```
NET"SW[0]"LOC=AA10 | IOSTANDARD=LVCMOS15;
NET"SW[1]"LOC=AB10 | IOSTANDARD=LVCMOS15;
NET"SW[2]"LOC=AA12 | IOSTANDARD=LVCMOS15;
NET"SW[3]"LOC=AA13 | IOSTANDARD=LVCMOS15;

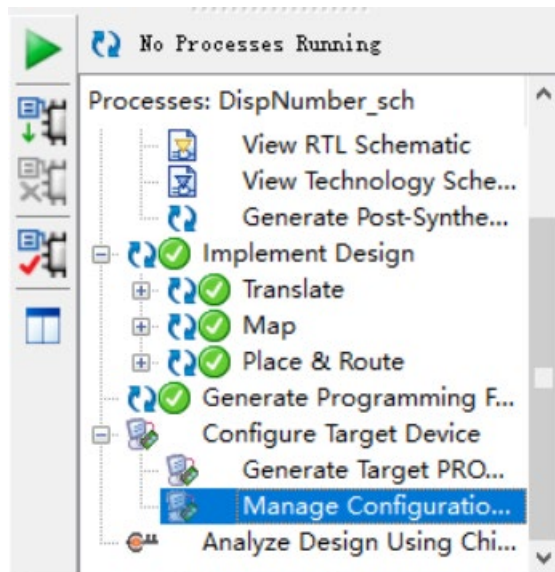
NET"SW[4]"LOC=Y13 | IOSTANDARD=LVCMOS15;
NET"SW[5]"LOC=Y12 | IOSTANDARD=LVCMOS15;
NET"SW[6]"LOC=AD11 | IOSTANDARD=LVCMOS15;
NET"SW[7]"LOC=AD10 | IOSTANDARD=LVCMOS15;

NET"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;
```

6. 引脚配置完毕后，将电脑连上实验平台的数据线，双击 Processes 窗口下的 Implement Design，成功后再双击 Generate Programming File，若成功，继续点击 Manage Configuration File。



7. 在 ISE IMPACT 平台上用类似之前实验的操作让 Program 在实验板上成功运行。

0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	2
0	0	1	1	0	0	0	0	1	1	0	3
0	1	0	0	1	0	0	1	1	0	0	4
0	1	0	1	0	1	0	0	1	0	0	5
0	1	1	0	0	1	0	0	0	0	0	6
0	1	1	1	0	0	0	1	1	1	1	7
1	0	0	0	0	0	0	0	0	0	0	8
1	0	0	1	0	0	0	0	1	0	0	9
1	0	1	0	0	0	0	1	0	0	0	A
1	0	1	1	1	1	0	0	0	0	0	b
1	1	0	0	0	1	1	0	0	0	1	C
1	1	0	1	1	0	0	0	0	1	0	d
1	1	1	0	0	1	1	0	0	0	0	E
1	1	1	1	0	1	1	1	0	0	0	F

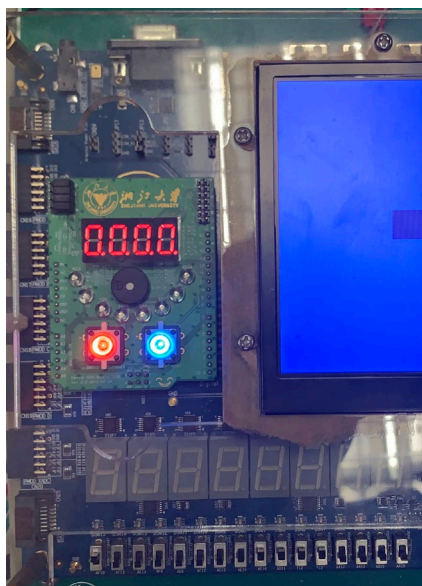
值得注意的是，由于本实验使用的是负逻辑电路，二极管再输入信号为0时发光，输入信号为1是不发光。在仿真测试电路的正确性的时候，要多加注意。

2.3 开关 SW(4) SW(5) SW(6) SW(7) 的真值表

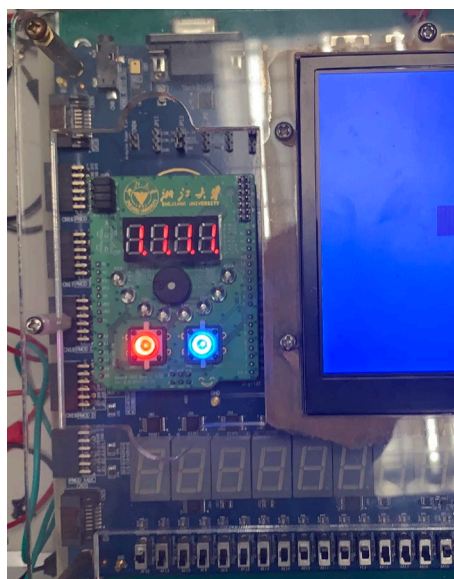
SW(7)	SW(6)	SW(5)	SW(4)	AD10	AD11	Y12	Y13
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	1
1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	1
1	1	0	0	1	1	0	0
1	1	0	1	1	1	0	1
1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	1

2.4 部分实验结果图片

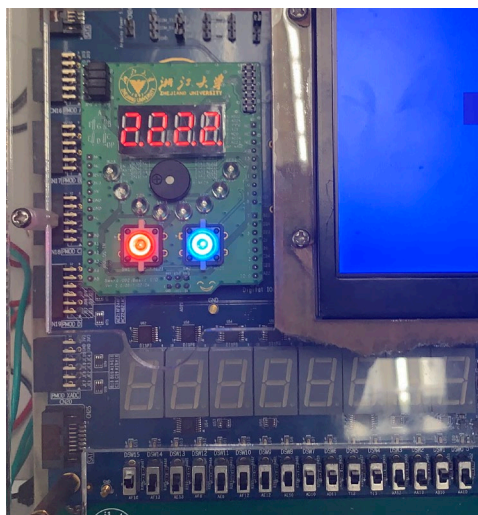
四个显示数字的使能开关打开（1），显示小数点的使能开关打开（1）。



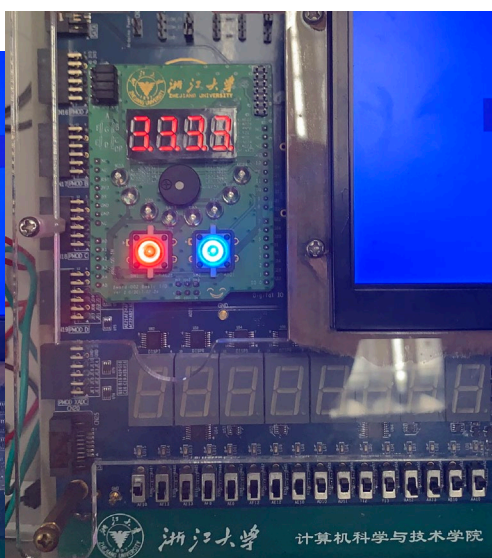
数字输入开关为 0 0 0 0



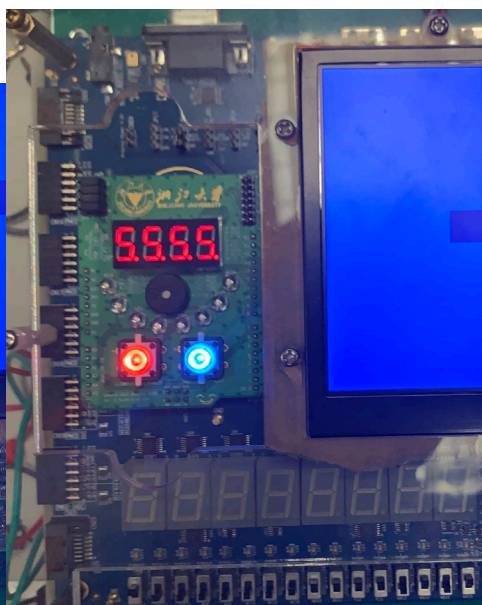
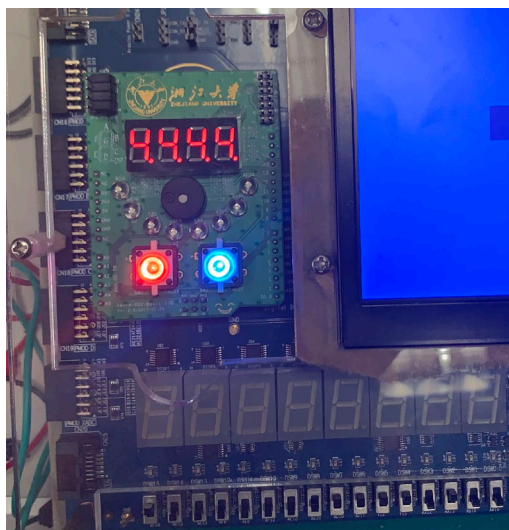
数字输入开关为 0 0 0 1



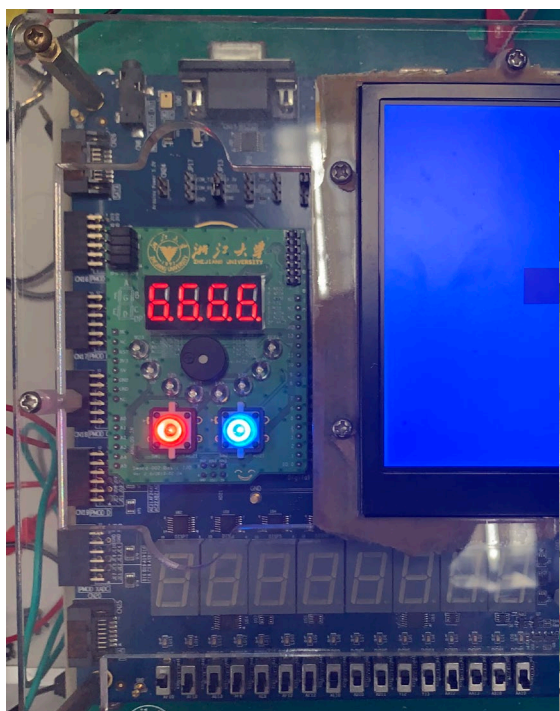
数字输入开关为 0 0 1 0



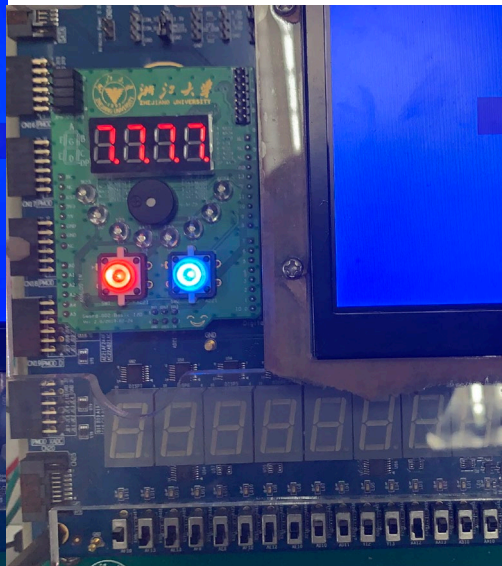
数字输入开关为 0 0 1 1



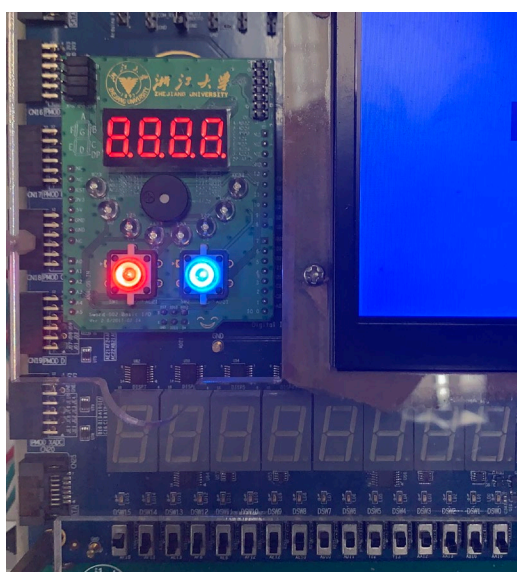
数字输入开关为 0 1 0 0



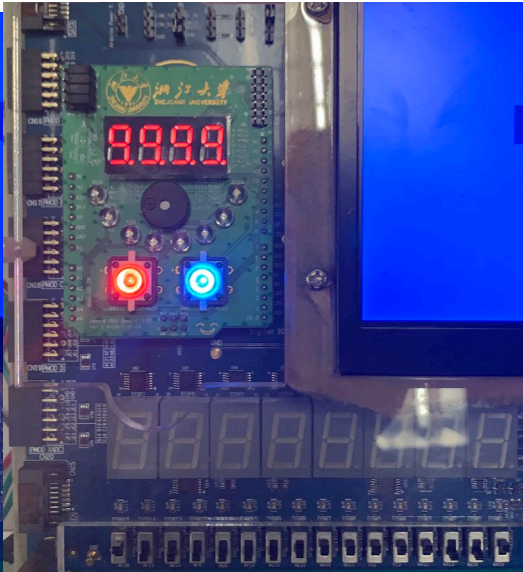
数字输入开关为 0 1 0 1



数字输入开关为 0 1 1 0

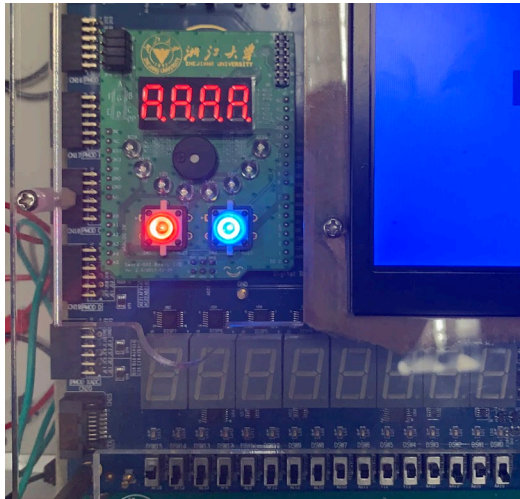


数字输入开关为 0 1 1 1



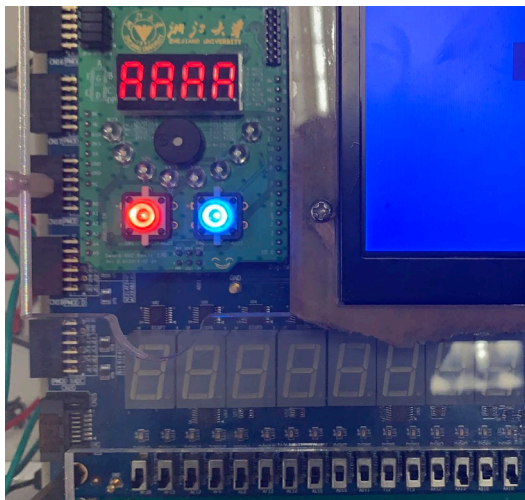
数字输入开关为 1 0 0 0

数字输入开关为 1 0 0 1

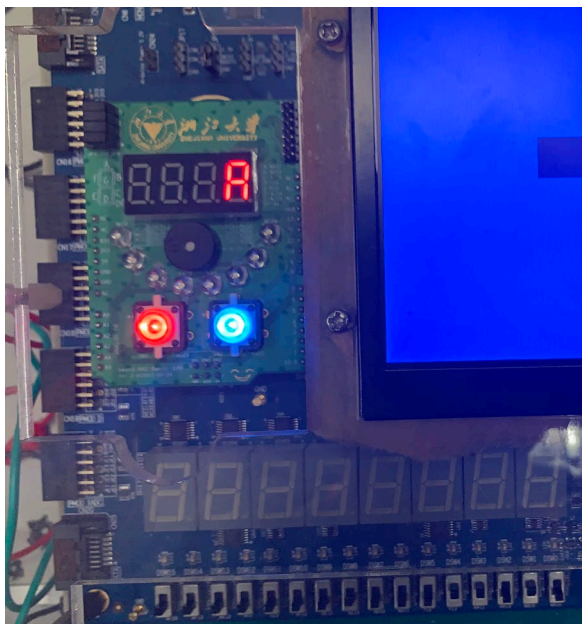


数字输入开关为 1 0 1 0

小数点使能开关关闭 (0)。数字输入开关为 1 0 1 0。四个显示数字的使能开关打开 (1)。



小数点使能开关关闭 (0)。数字输入开关为 1 0 1 0。四个显示数字的使能开关为 0 0 0 1。



经过检验，所有结果都符合预期设想，逻辑电路的正确性在数字逻辑电路板上成功得到了验证。

三、讨论、心得

实验中，想到了之后检验电路的时候如果引脚对应得不规整，将会加大模拟的困难，故在引脚配置的时候，把较大的精力都投入在引脚的设计上。

```
NET"SW[0]"LOC=AA10 | IOSTANDARD=LVCMOS15;
NET"SW[1]"LOC=AB10 | IOSTANDARD=LVCMOS15;
NET"SW[2]"LOC=AA12 | IOSTANDARD=LVCMOS15;
NET"SW[3]"LOC=AA13 | 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"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;
```

经过考虑，选择把 SW(0), SW(1), SW(2), SW(3)的引脚分别配置对应数字逻辑电路板上的 AA10, AB10, AA12, AA13, 也就是实验电路板上从最右边、自右向左的四个开关。这样的话，从左向右，四个开关将分别对应 D3, D2, D1, D0, 十分规整，非常有利于后期的实验模拟与测试。两个使能 BTN(1)和 BTN(0)的引脚配置的是 AF10 和 AF13, 是逻辑电路板上最左边的两个开关，它们分别控制所有数字发光管的亮暗和所有小数点发光单元的亮暗。这样，后期的模拟就十分方便，显示出的数字的二进制就仿佛与相应的开关组合所表示的数组相同。例如，如果要表示数字“7”，其二进制表示为“0111”，而开关的真值，从左到右，也正好是“0111”。

本实验，我还深刻地意识到仿真的重要作用。在之前的实验里，由于相对比较顺利，也没有出现连线错误的情况，仿真过程一直感觉意义不大。而这一次，由于实验电路图十分复杂，

联想过程中一不小心就会出错，而且还不易发现，这个时候学会通过观察仿真电路图查找错误就显得十分高效。实验使用的是负逻辑电路，意味着二极管再输入为“0”时才会发光，依照这一点去检验 MyMC14495 模块的仿真波形，就能很快地找出错误，并加以改正。