

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:	王晓宇
学 院:	竺可桢学院
专 业:	计算机
邮 箱:	1657946908@qq.com
QQ 号:	1657946908
电 话:	19550222634
指导教师:	洪奇军
报告日期:	2023 年 10 月 19 日

浙江大学实验报告

课程名称：____计算机逻辑设计基础____ 实验类型：____综合____

实验项目名称：____变量译码器设计与应用____

学生姓名：____王晓宇____ 学号：____3220104364____ 同组学生姓名：____

实验地点：____紫金港东四 509 室____ 实验日期：____2023____ 年 ____10____ 月 ____19____ 日

一、操作方法与实验步骤

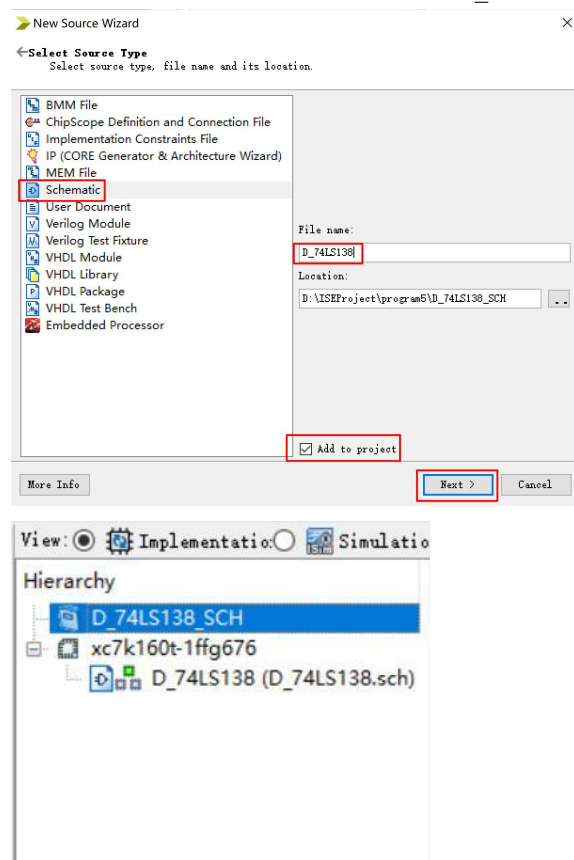
（一）原理图设计实现 74LS138 译码器模块

1、新建工程

新建工程，工程名命名为 D_74LS138_SCH。

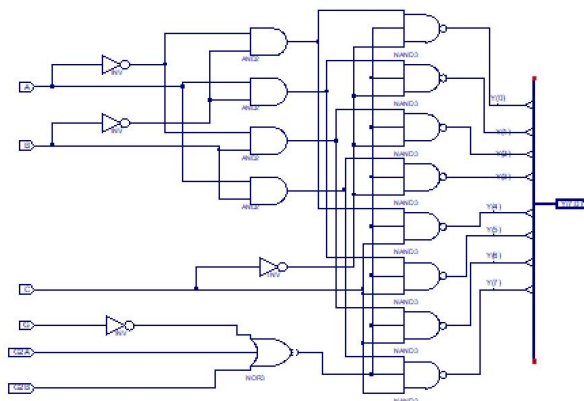
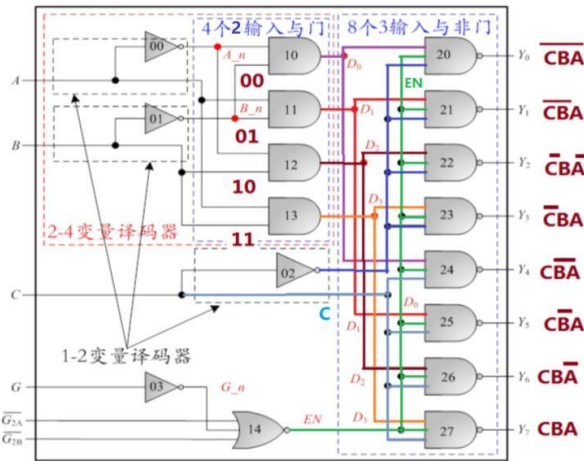
2、新建 Schematic 源文件

新建 Schematic 源文件，文件命名为 D_74LS138。新建好的工程和源文件如下：



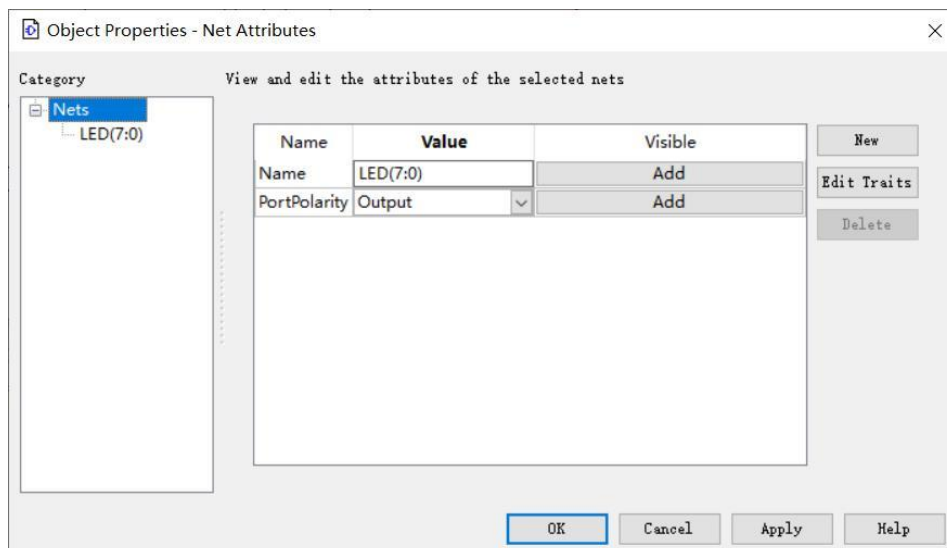
3、绘制 74LS138 译码器原理图

（1）74LS138 译码器原理



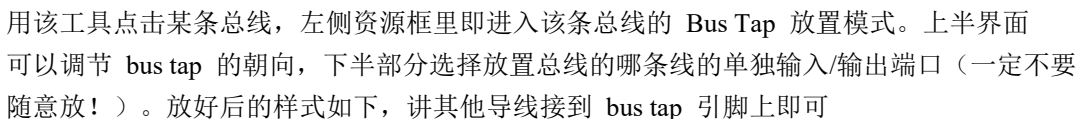
(2) 总线结构的绘制

- ①先用 ADD_WIRE 工具画一条线。
- ②双击该条电线，编辑这条线的名称，如下：



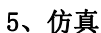
其中，Name 的格式为 name(n:0)，“name”为这条线的名字，“(n:0)”表示这条线为 (n+1) 条单条导线的总线，单条导线的编号依次为 0~n。

- ③总线数据流的输入输出



需要注意做好重命名，这样当此设备作为一个器件投入使用时，各个输入输出端对应什么数据才更加明确。

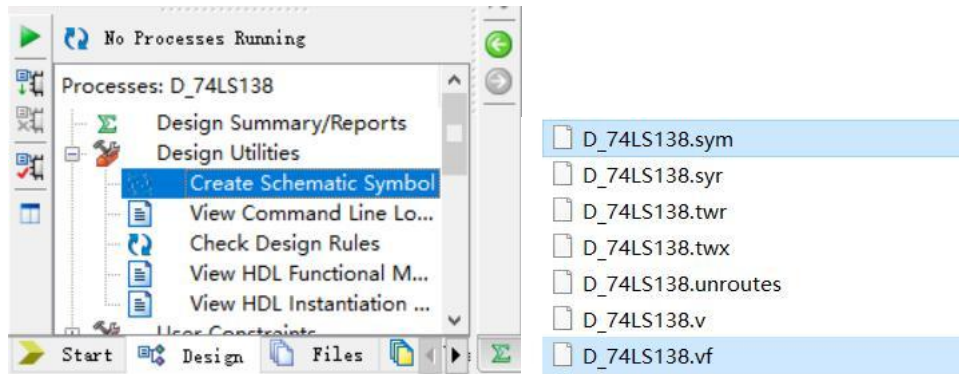
依次对该 schematic 文件运行 Check Design Rules、Synthesize。运行 View HDL Functional Module，查看并学习 Verilog 代码。



Name	Value	0 ns	100 ns	200 ns	300 ns	400 ns	500 ns
G	1	[Timing diagram showing signal G as a high pulse from 0 ns to approximately 100 ns]					
G2A	0	[Timing diagram showing signal G2A as a low pulse from 0 ns to approximately 100 ns]					
G2B	0	[Timing diagram showing signal G2B as a low pulse from 0 ns to approximately 100 ns]					
A	0	[Timing diagram showing signal A as a low pulse from 0 ns to approximately 100 ns]					
B	0	[Timing diagram showing signal B as a low pulse from 0 ns to approximately 100 ns]					
C	0	[Timing diagram showing signal C as a low pulse from 0 ns to approximately 100 ns]					
Y[7:0]	11111110	[Timing diagram showing signal Y[7:0] as a high pulse from 0 ns to approximately 100 ns]					
[7]	1	[Timing diagram showing signal [7] as a high pulse from 0 ns to approximately 100 ns]					
[6]	1	[Timing diagram showing signal [6] as a high pulse from 0 ns to approximately 100 ns]					
[5]	1	[Timing diagram showing signal [5] as a high pulse from 0 ns to approximately 100 ns]					
[4]	1	[Timing diagram showing signal [4] as a high pulse from 0 ns to approximately 100 ns]					
[3]	1	[Timing diagram showing signal [3] as a high pulse from 0 ns to approximately 100 ns]					
[2]	1	[Timing diagram showing signal [2] as a high pulse from 0 ns to approximately 100 ns]					
[1]	1	[Timing diagram showing signal [1] as a high pulse from 0 ns to approximately 100 ns]					
[0]	0	[Timing diagram showing signal [0] as a low pulse from 0 ns to approximately 100 ns]					
[31:0]	XXXXXXXXXX	[Timing diagram showing signal [31:0] as a high pulse from 0 ns to approximately 100 ns]					

如果我们要把我们的电路封装为一个元件，以后直接用，不用再从逻辑门开始画起，就需要生成对应元件。

运行 Create Schematic Symbol，就能产生该元件的.sym 文件。同时，在上述的 View HDL Functional Module 这一步，我们生成了该元件.vf 文件。如果在其他工程中使用该元件，只需要把.sym 和.vf 文件加入到其他工程。



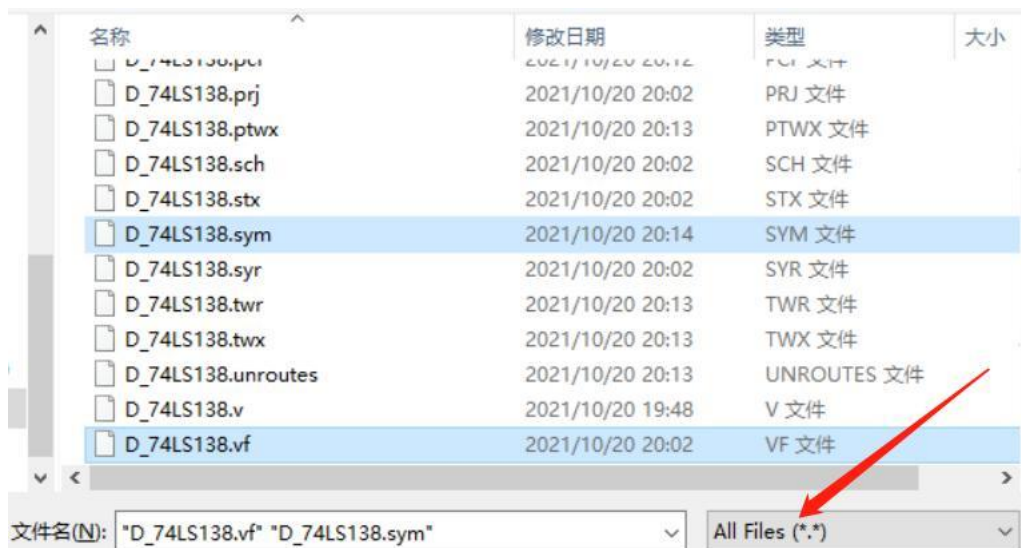
(二) 验证 74LS138 译码器的逻辑功能

1、新建工程 D_74LS138_Test。

2、新建 Schematic 文件“D74LS138TEST”。

3、在资源区域右键，选择“Add Copy of Source”，将第一个工程目录下的 D74LS138.vf 和 D74LS138.sym 文件加入到工程。

注：（1）在文件目录下不能直接找到.vf 和.sym 后缀的文件，需要将文件类型切换到 All File。

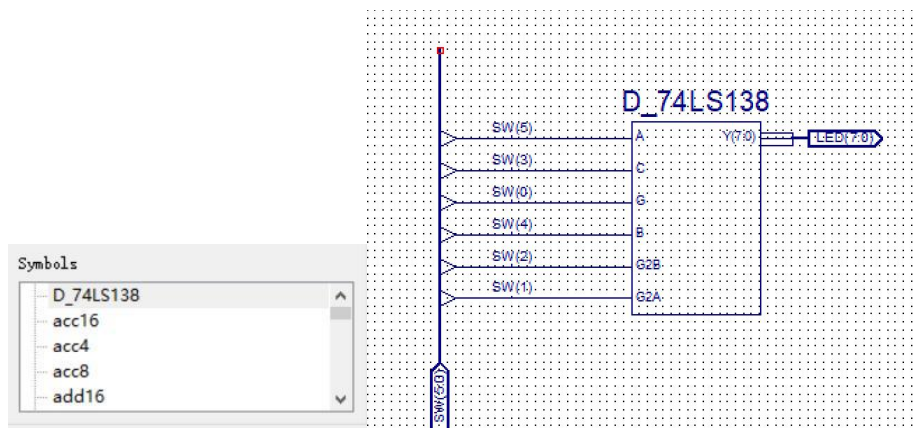


（2）.sym 文件是运行“Create Schematic Symbol”后生成的元器件符号封装文件。

.vf 文件是运行“View HDL Functional Model”后生成的 schematic 文件的 verilog 代码文件，描述元器件的内部结构，控制逻辑功能的实现。

4、利用 D_74LS138 元件绘图。

导入元件后，即可在 symbols 中找到自建的元件。

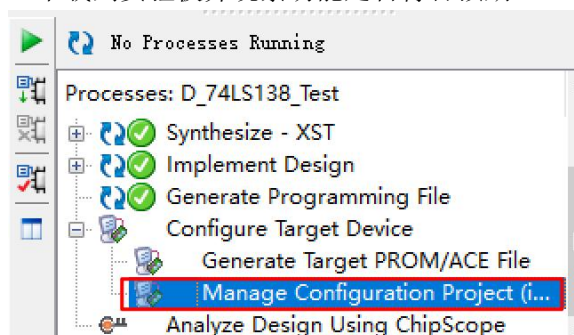


(4) 引脚约束文件

① 新建引脚约束文件

```
1 NET"LED[0]"LOC=W23 | IOSTANDARD=LVCMOS33;
2 NET"LED[1]"LOC=AB26 | IOSTANDARD=LVCMOS33;
3 NET"LED[2]"LOC=Y25 | IOSTANDARD=LVCMOS33;
4 NET"LED[3]"LOC=AA23 | IOSTANDARD=LVCMOS33;
5 NET"LED[4]"LOC=Y23 | IOSTANDARD=LVCMOS33;
6 NET"LED[5]"LOC=Y22 | IOSTANDARD=LVCMOS33;
7 NET"LED[6]"LOC=AE21 | IOSTANDARD=LVCMOS33;
8 NET"LED[7]"LOC=AF24 | IOSTANDARD=LVCMOS33;
9
10 NET"SW[0]"LOC=Y12 | IOSTANDARD=LVCMOS15;
11 NET"SW[1]"LOC=Y13 | IOSTANDARD=LVCMOS15;
12 NET"SW[2]"LOC=AA12 | IOSTANDARD=LVCMOS15;
13 NET"SW[3]"LOC=AA13 | IOSTANDARD=LVCMOS15;
14 NET"SW[4]"LOC=AB10 | IOSTANDARD=LVCMOS15;
15 NET"SW[5]"LOC=AA10 | IOSTANDARD=LVCMOS15;
```

(5) 下载到实验板并观察功能是否符合预期



(三) 实现楼道灯的控制

1、新建工程 LampCtrl。

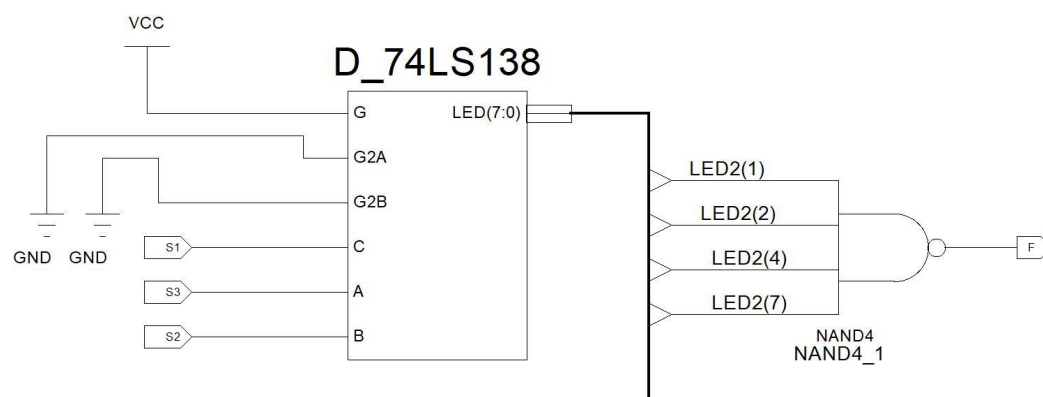
2、新建 Schematic 文件 Lamp。

3、将 D_74LS138 译码器的.vf 和.sym 文件添加到此工程中。

4、绘制楼道灯的原理图

1 用 VCC 表示，0 用 GND 表示。

利用此电路可实现：当 ABC 中任意一个的高低电平状态改变时，输出信号都会改变，从而实现楼道灯（房间灯）按任意一个开关都能改变灯的暗灭状态的功能。



5、楼道灯的仿真

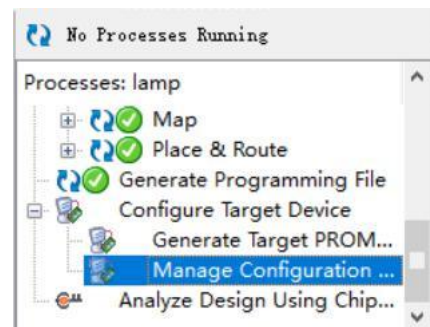
新建仿真激励文件 lamp_sim.v，加入信号灯的仿真代码，进行仿真。

6、引脚约束

对楼道灯和开关进行引脚约束：

```
1 NET "S1" LOC=AA10 | IOSTANDARD=LVCMOS15;  
2 NET "S2" LOC=AB10 | IOSTANDARD=LVCMOS15;  
3 NET "S3" LOC=AA13 | IOSTANDARD=LVCMOS15;  
4 NET "F" LOC=AF24 | IOSTANDARD=LVCMOS33 ;#D8  
5
```

7、生成 bit 文件，将 bit 文件下载到 sword 板上，进行功能测试。

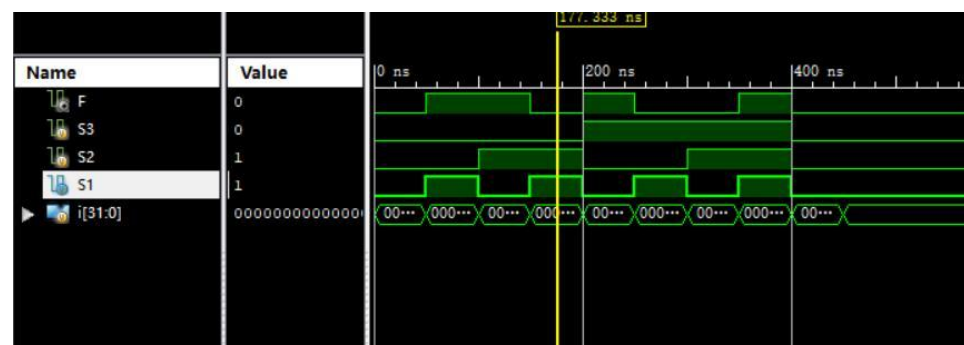


二、实验结果与分析

仿真：



$\{G, G2A, G2B\} == 000, F = 0$



$\{G, G2A, G2B\} == 110, F = 1$

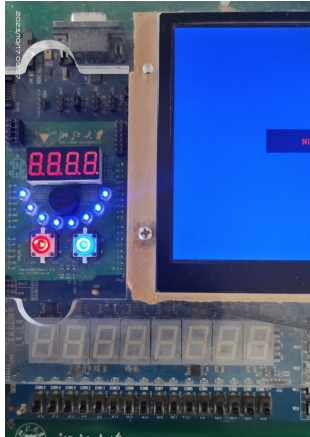
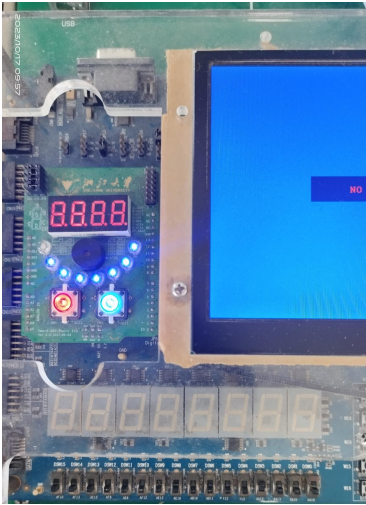
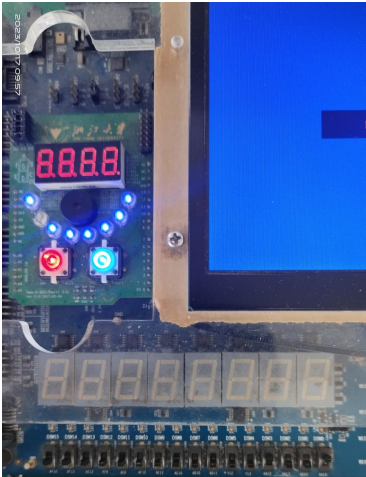


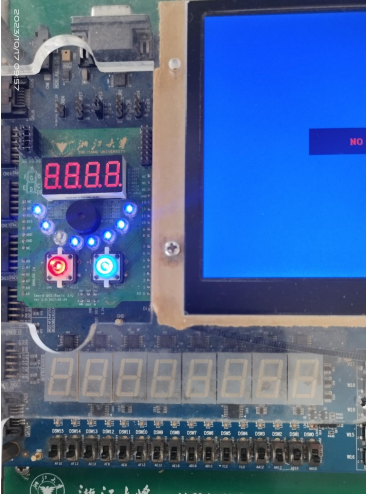
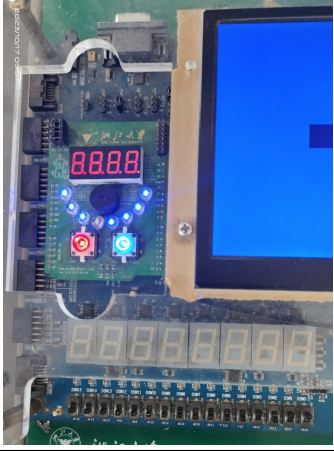
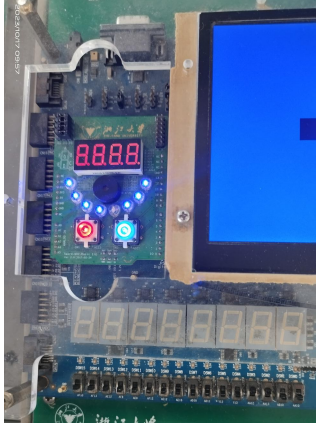
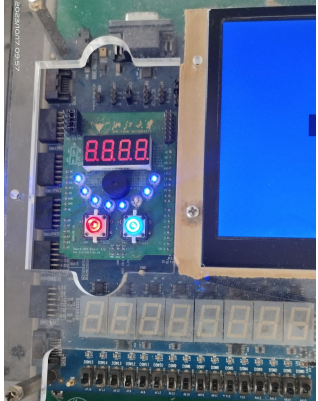
$\{G,G2A,G2B\}==101, F=0$


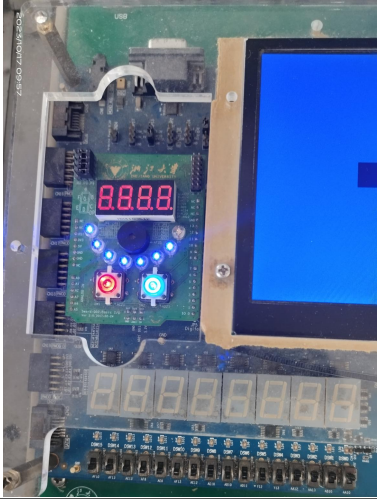


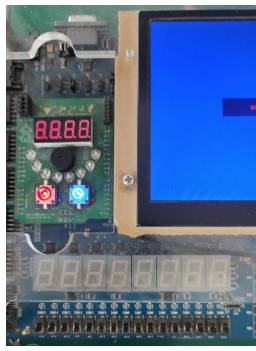
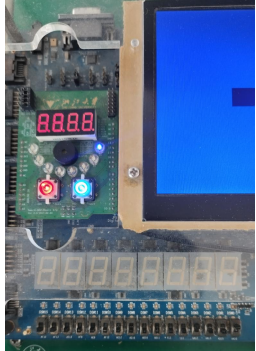
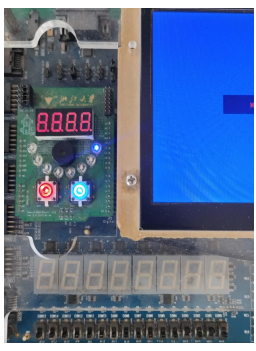
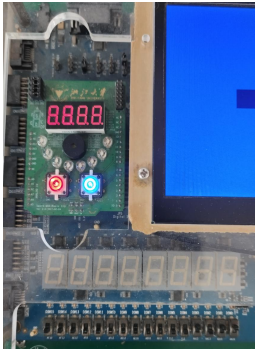
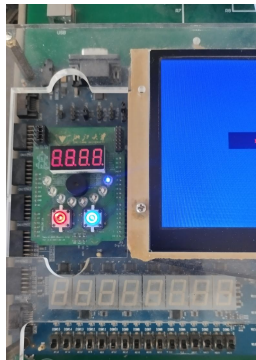
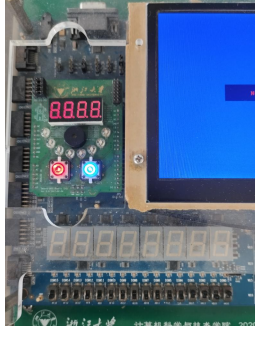
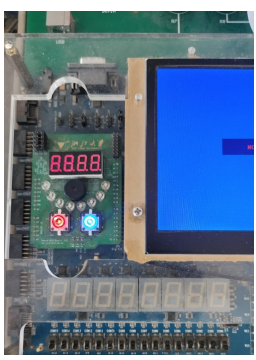
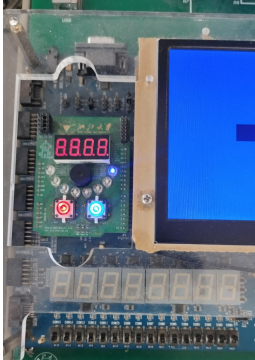
$\{G,G2A,G2B\}==111, F=1$

$\{G,G2A,G2B\}$	$\{C,B,A\}$	实验结果
000	000	

011	000	
100	000	
100	001	

100	010	
100	011	
100	100	
100	101	

100	110	
100	111	

			
000	001	010	011
			
100	101	110	111

三、讨论、心得

这是第二次上板验证，在学习过程中发现一种新的构建电路的方式，即通过先写好一个元器件文件生成.sym 文件直接 **add copy of source** 来用到其他文件里面，这样子比较简化电路，缺点是第二个使用的人不容易知道该器件的作用，需另加说明文档；另一个方面，我发现第三个 **project** 可以不用将所有输出接到输出端，经过逻辑验证后的确不需要另外的引脚来输出，这样子设计电路可以减少导线数量和通信总数，私以为可以提高元件工作效率。