**实验五——变量译码器设计与应用实验报告**

姓名： 王祚滨 专业： 信息安全 学号： 3180104933

课程名称： 逻辑与计算机设计基础实验 同组学生姓名： 王国朝、赵卿云

指导老师： 洪奇军 实验地点： 浙江大学紫金港校区东四教学楼509实验日期：2019年 10 月 16 日

# 一、实验目的和要求

1.掌握变量译码器的的逻辑构成和逻辑功能。

2. 用变量译码器实现组合函数

3. 掌握变量译码器的典型应用（地址译码的具体方法）

4. 了解存储器编址的概念

5. 采用原理图设计电路模块

6. 进一步熟悉ISE平台及下载实验平台

**二、实验内容和原理**

2.1实验内容

2.1.1原理图设计实现74LS138译码器模块

2.1.2 用74LS138译码器实现楼道灯控制

2.2实验原理：

2.2.1译码器原理

译码器是将一种输入编码转换成另一种编码的电路，即将给定的代码进行“翻译”并转换成指定的状态或输出信号（脉冲或电平）。

译码可分为：变量译码、显示译码。

变量译码一般是将一种较少位输入变为较多位输出的器件，如2n译码和8421BCD码译码。

显示译码主要进行2进制数显示成10进制或16进制数的转换，可分为驱动LED和LCD两类。

变量译码器是一个将n个输入变为2n个最小项输出的多输出端的组合逻辑电路。n通常在2~64之间。



2.2.2变量译码器—74LS138介绍

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **输入** | | **译码器输出**  **（低电平有效）** | | | | | | | |
| **使能** | **变量** |
| **GG2AG2B** | **CBA** | **Y0** | **Y1** | **Y2** | **Y3** | **Y4** | **Y5** | **Y6** | **Y7** |
| ×11 | ××× | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0×× | ××× | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 100 | 000 | **0** | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 100 | 001 | 1 | **0** | 1 | 1 | 1 | 1 | 1 | 1 |
| 100 | 010 | 1 | 1 | **0** | 1 | 1 | 1 | 1 | 1 |
| 100 | 011 | 1 | 1 | 1 | **0** | 1 | 1 | 1 | 1 |
| 100 | 100 | 1 | 1 | 1 | 1 | **0** | 1 | 1 | 1 |
| 100 | 101 | 1 | 1 | 1 | 1 | 1 | **0** | 1 | 1 |
| 100 | 110 | 1 | 1 | 1 | 1 | 1 | 1 | **0** | 1 |
| 100 | 111 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | **0** |

图2.2.2-1 74LS138变量译码器功能表



图2.2.2-2 74LS138变量译码器引脚

图2.2.2-3 74LS138原理图（带3个使能端的3-8译码器的逻辑结构由三级门电路构成，输出低电平有效）

2.2.3 用变量译码器实现组合函数

变量译码器的输出对应所有输入变量的最小项组合，如果将函数转换成最小项和的形式，则可以用变量译码器实现函数的组合电路:

F = S3(bar)S2(bar)S1+S3(bar)S2S1(bar)+S3S2(bar)S1(bar)+S3S2S1

F = 001 + 010 + 100 +111



图2.2.3-1 用74LS138译码器实现楼道灯控制原理图

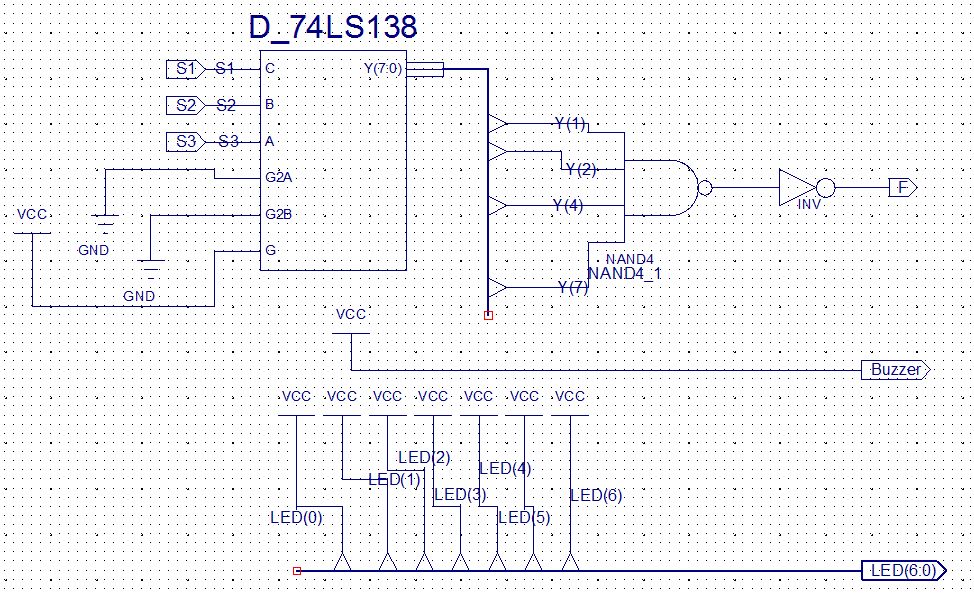


图2.2.3-2 用74LS138译码器实现楼道灯控制完整原理图

# 三、主要仪器设备

1.装有Xilinx ISE 14.7的计算机 1台

2.SWORD开发板

# 四、操作方法与实验步骤

4.1 原理图设计实现74LS138译码器模块

4.1.1 新建工程，工程名称用D\_74LS138\_SCH。（第一个工程）

4.1.2 新建Schematic源文件，文件名称用D\_74LS138。

4.1.3 原理图方式进行设计，得到结果如下。

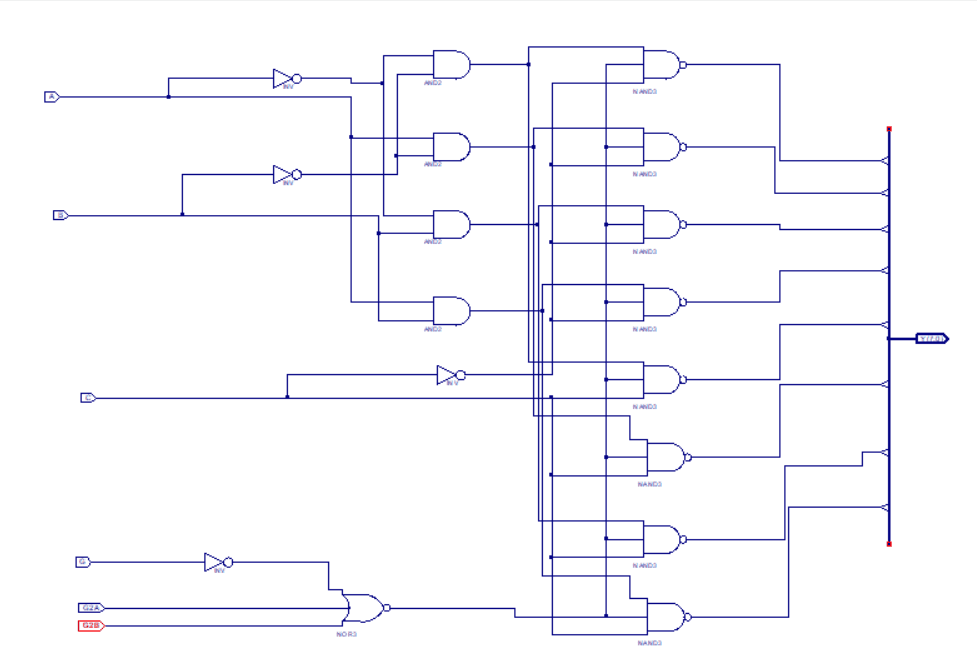


图4.1.3-1 74LS138原理图

4.1.4 Check Design Rules，检查错误

4.1.5 View HDL Functional Model，查看并学习Verilog HDL代码，得到HDL代码如下

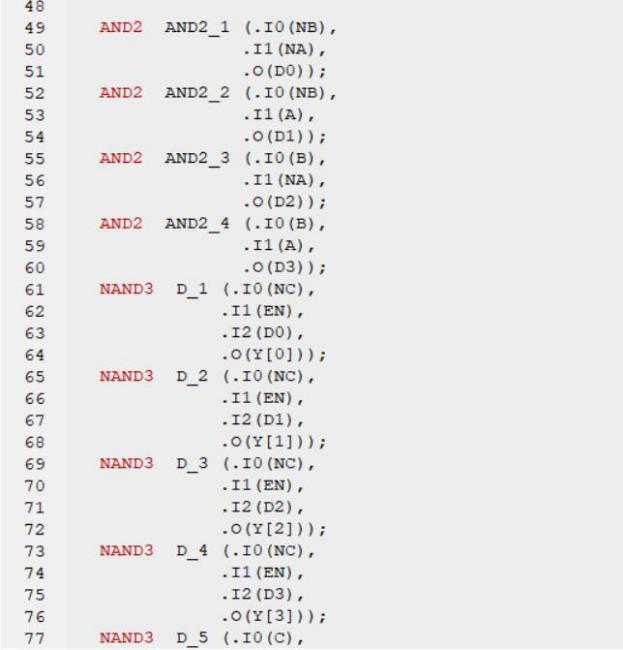
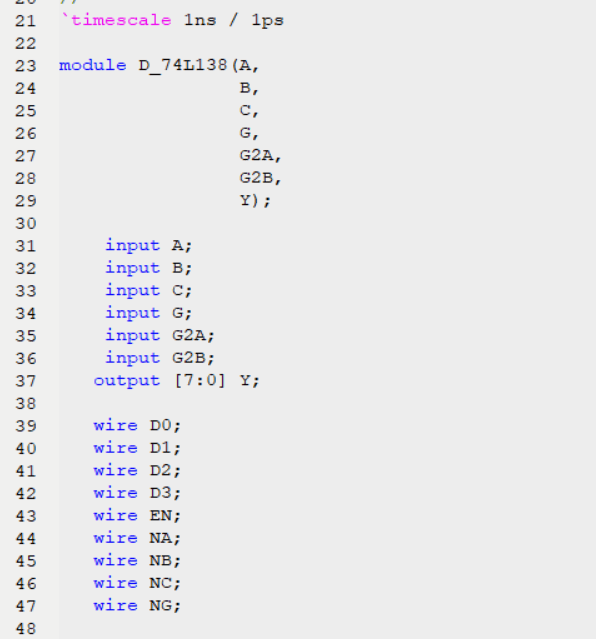
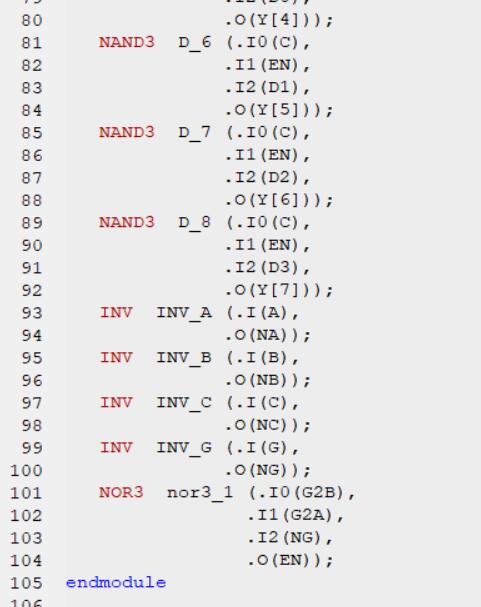
 

图4.1.5 HDL代码图

4.1.6 仿真

加入激励代码

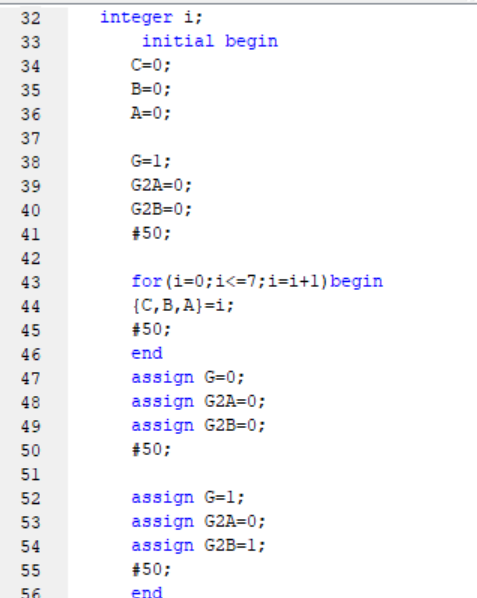


图4.1.6-1 激励代码

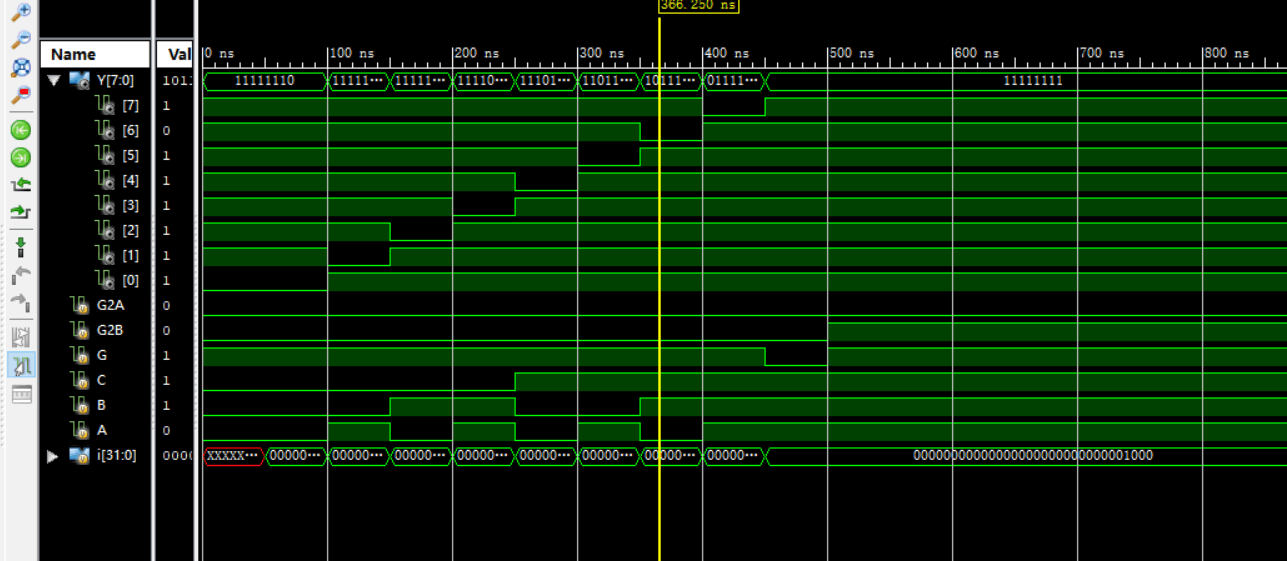


图4.1.6-2 仿真图

4.1.7生成逻辑符号图和VF文件

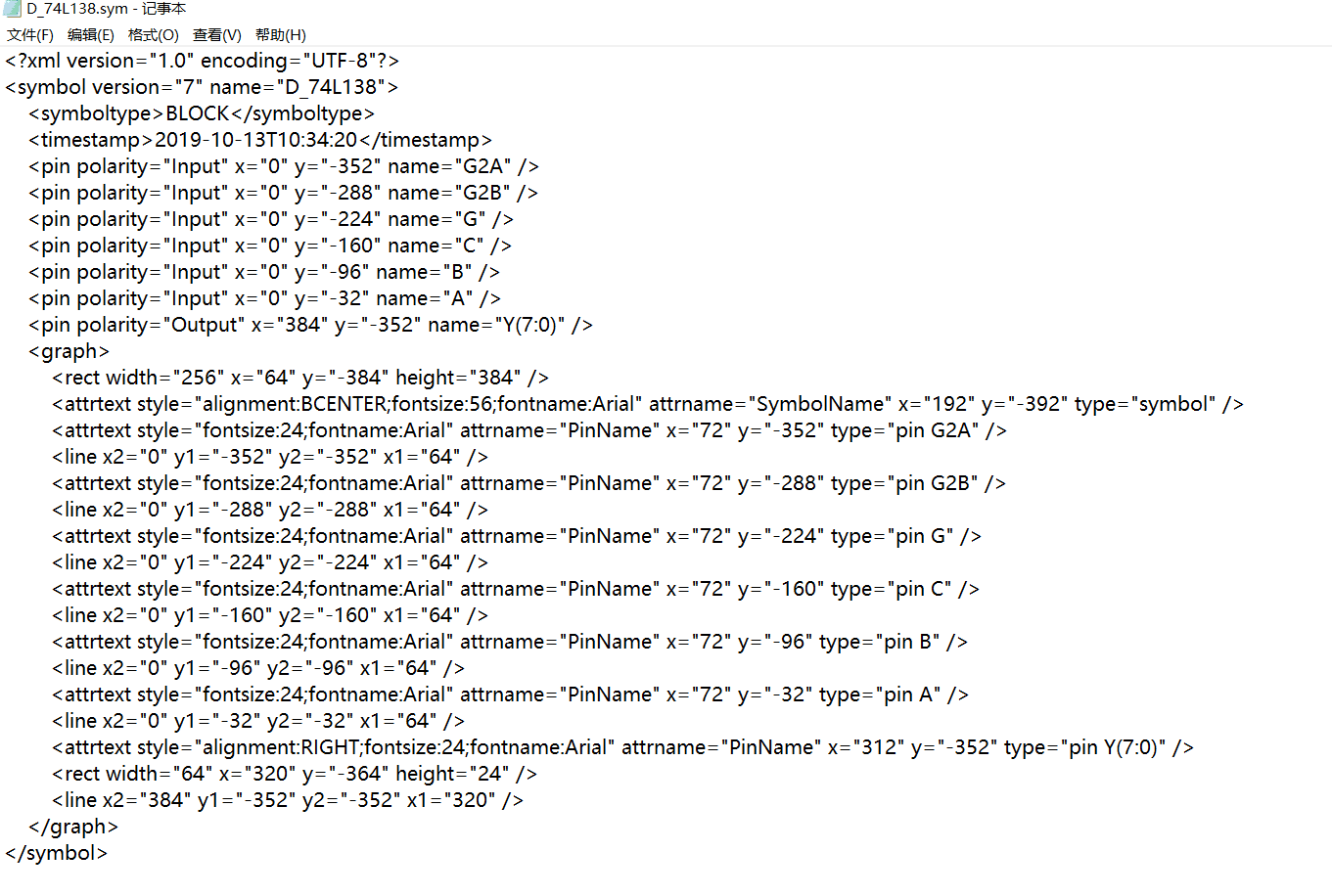


图4.1.7-1 记事本打开.sym文件

4.2验证D\_74LS138

4.2.1 新建工程 D\_74LS138\_Test（第二个工程）

4.2.2 新建Schematic文件“D\_74LS138\_Test”，复制D\_74LS138.sym和.vf到工程目录。

4.2.3 画出原理图

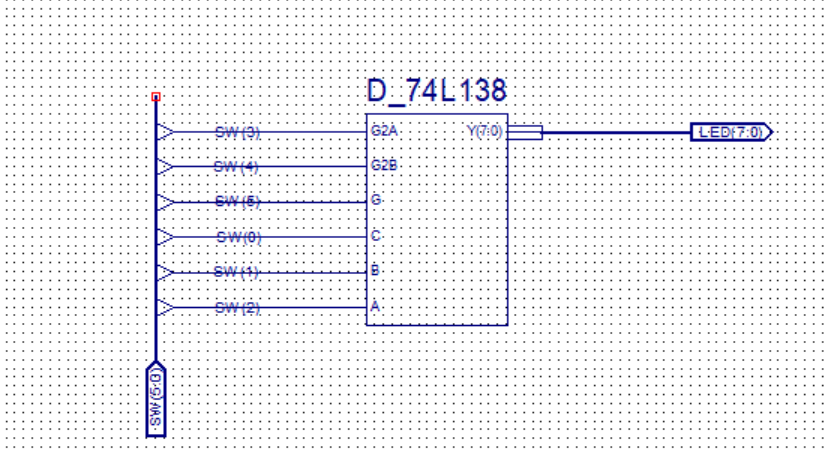


图4.2.3 D\_74LS138\_Test 原理图

4.2.4 UCF引脚定义

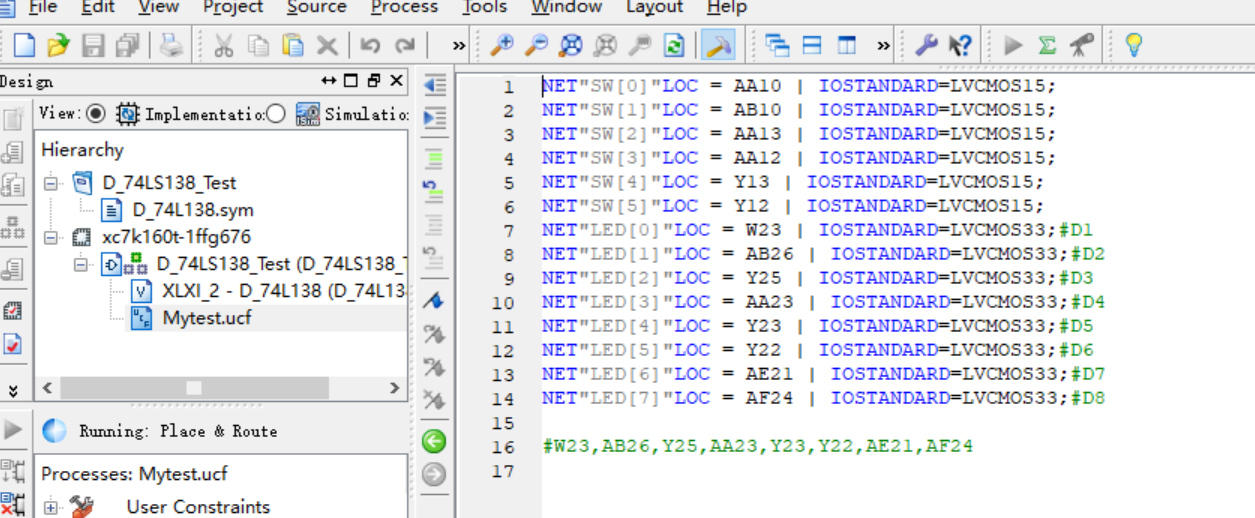


图4.2.4 ucf引脚定义

4.2.5 下载到实验板，验证功能

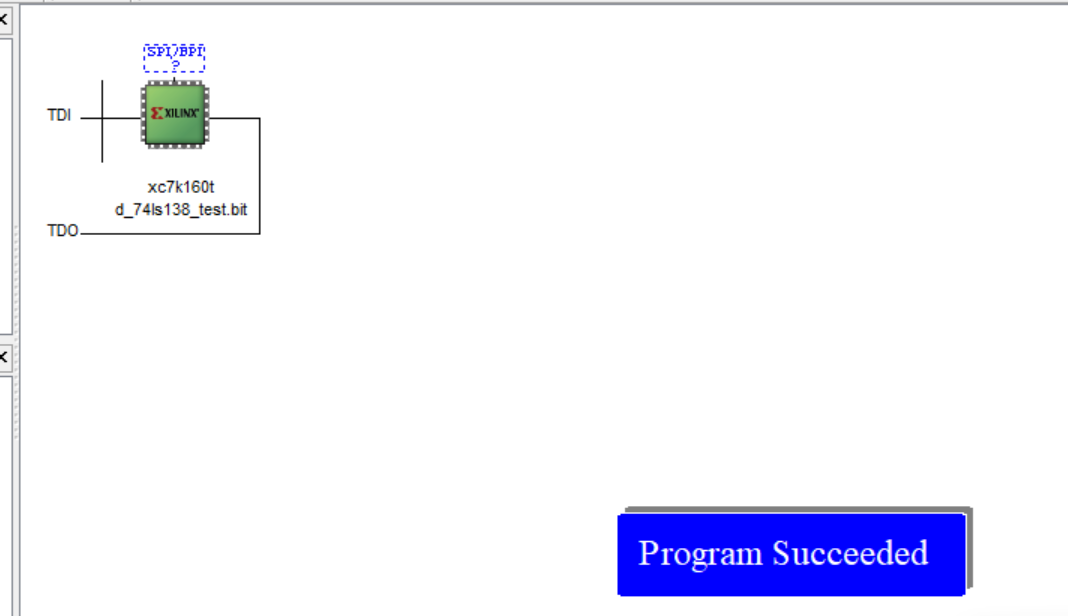


图4.2.5 program

4.3实现楼道灯控制

4.3.1 新建工程LampCtrl138,创建sch原理图

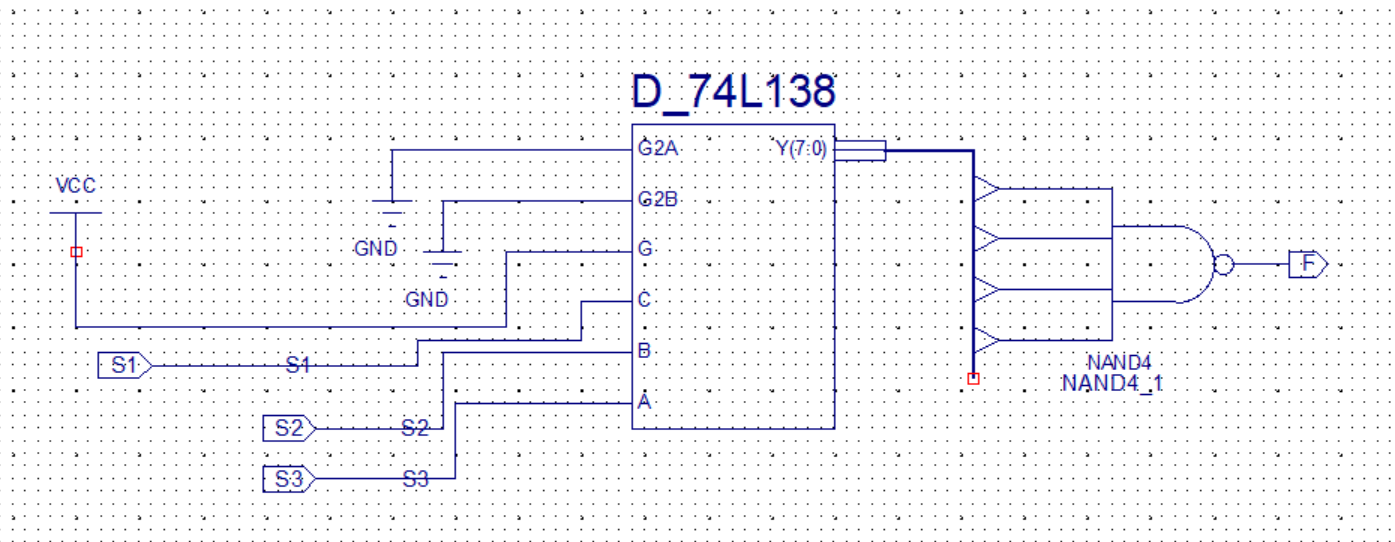


图4.3.1 原理图

4.3.2 仿真

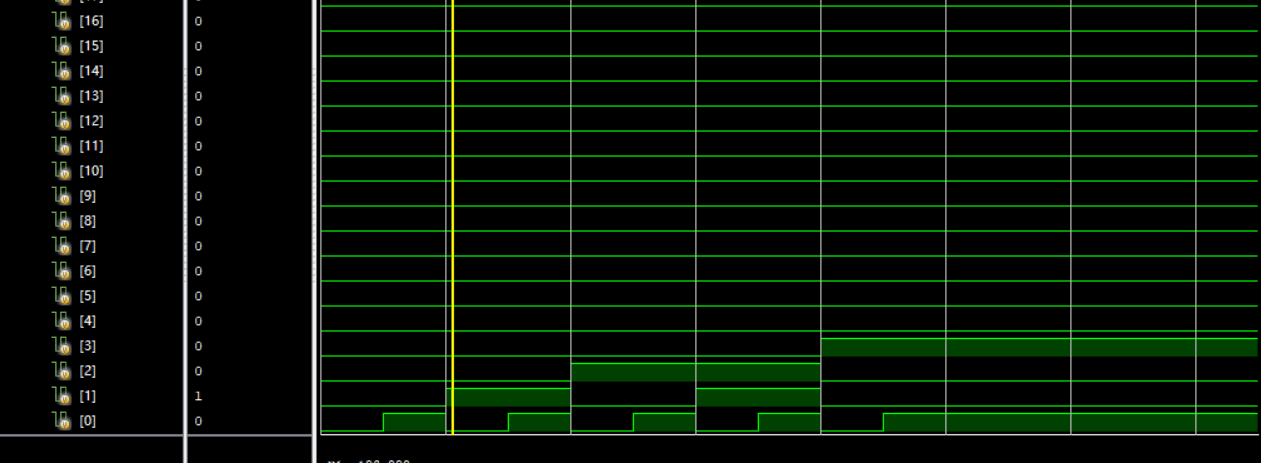


图4.3.2-1 仿真图(1)

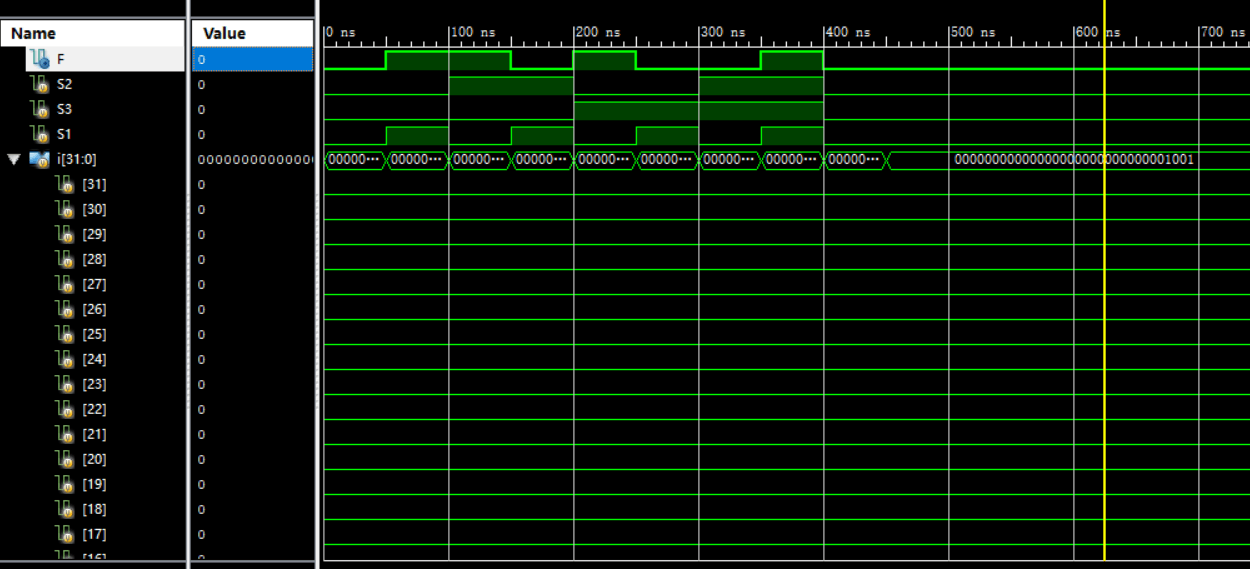


图4.3.2-2 仿真图(2)

4.3.3建立引脚约束

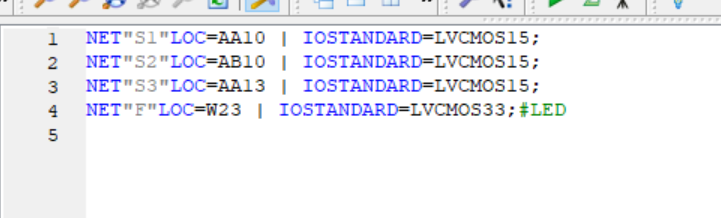
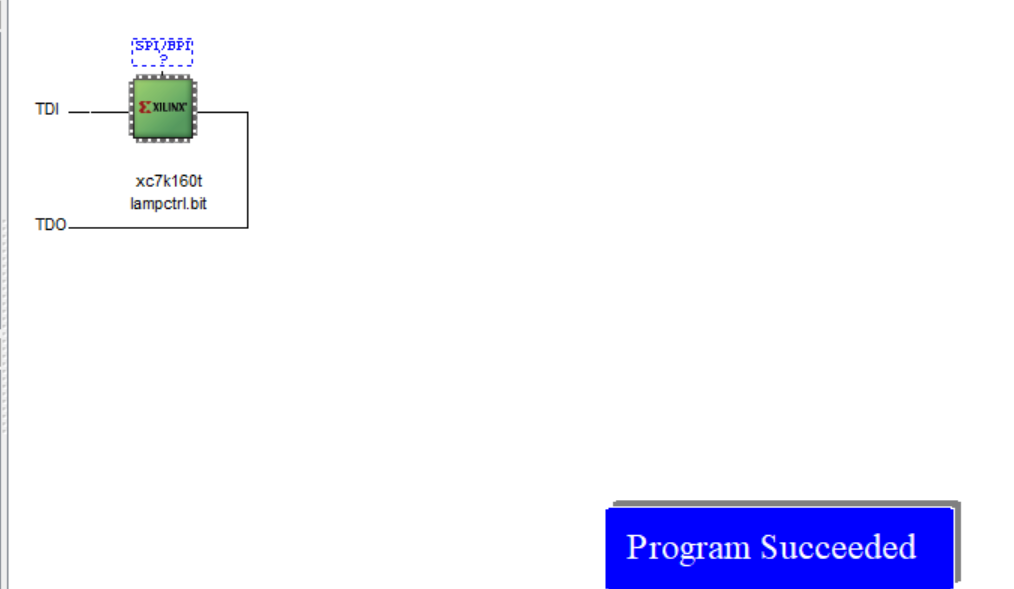


图4.3.3 引脚定义

4.3.4 下载验证



# 实验结果与分析

5.1.1变量译码器-74LS138的实现

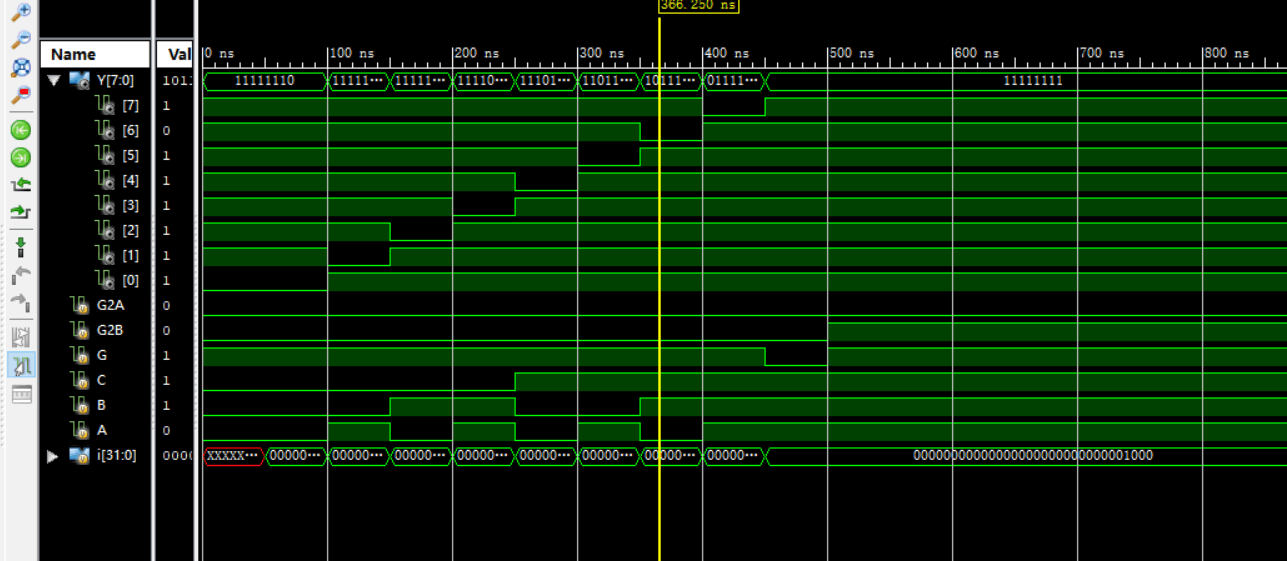


图5.1.1 译码器仿真图

5.1.2 仿真图分析

该译码器是将CBA三个输入的二进制数翻译为对应的八进制数，输出低电平有效。从图中可以观察到，CBA从000开始变化，每次加一，依次为000,001,010,011,100,101,110,111，对应到Y[7:0]这八个数中的一个，被对应到的数电平变为0，所以Y[0],Y[1],Y[2], Y[3],Y[4],Y[5],Y[6],Y[7]依次变为0，其余时刻均为1。

5.2.1 验证D\_74LS138

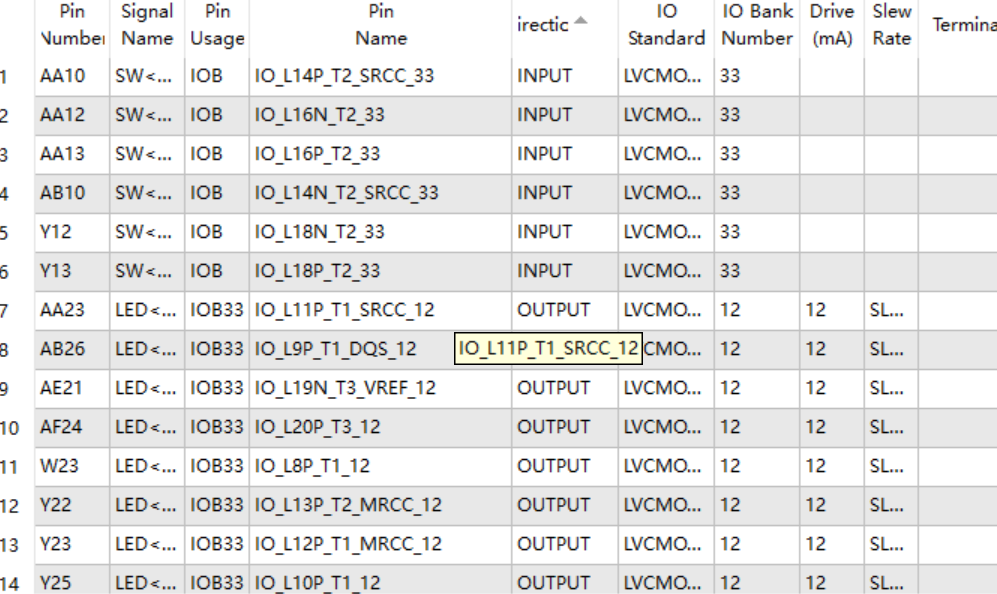


图5.2.1-1 pinout report

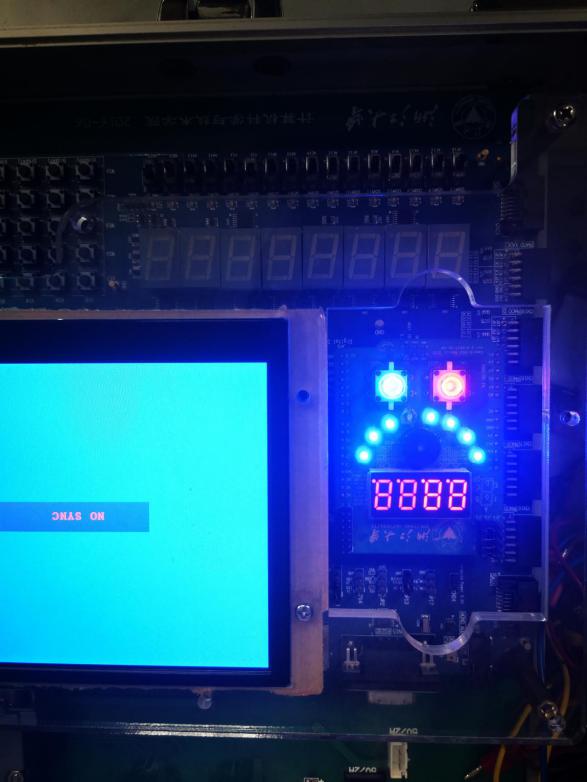
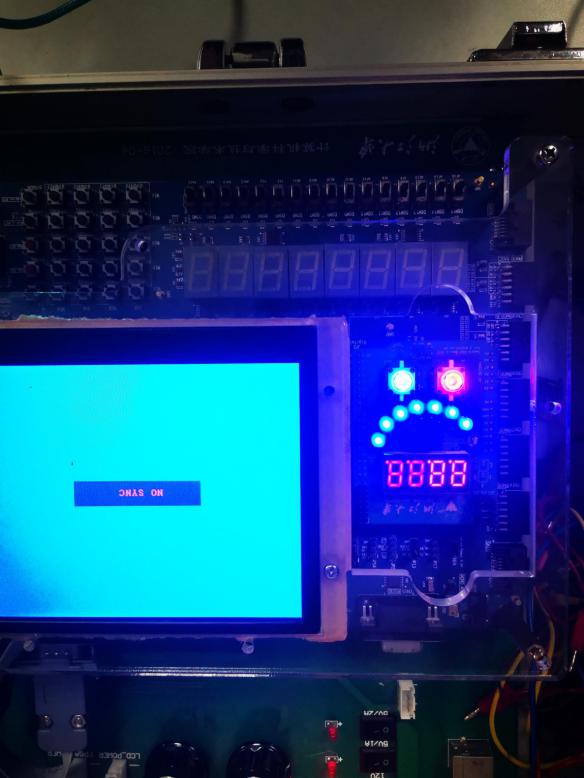


图5.2.1-2 验证结果

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 输入 | | 译码器输出  （低电平有效） | | | | | | | |
| 使能 | 变量 |
| GG2AG2B | CBA | Y0 | Y1 | Y2 | Y3 | Y4 | Y5 | Y6 | Y7 |
| ×11 | ××× | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0×× | ××× | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 100 | 000 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 100 | 001 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 100 | 010 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 100 | 011 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 100 | 100 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 100 | 101 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 100 | 110 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 100 | 111 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

5.2.1-3 真值表

使能为1，0，0时，得到结果符合真值表，验证成功

5.3.1 实现楼道灯控制

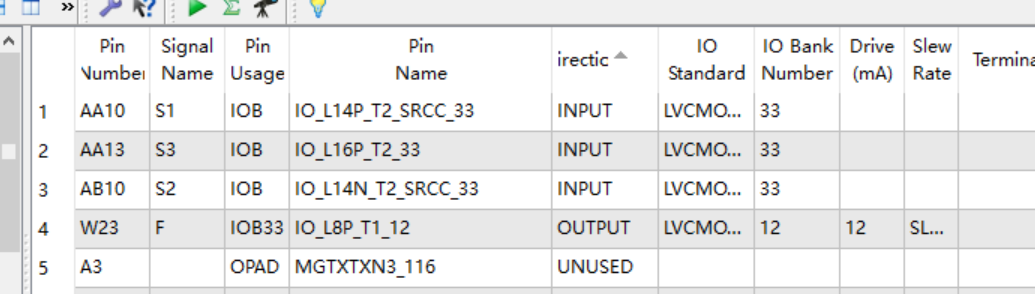


图5.3.1-1 pinout report

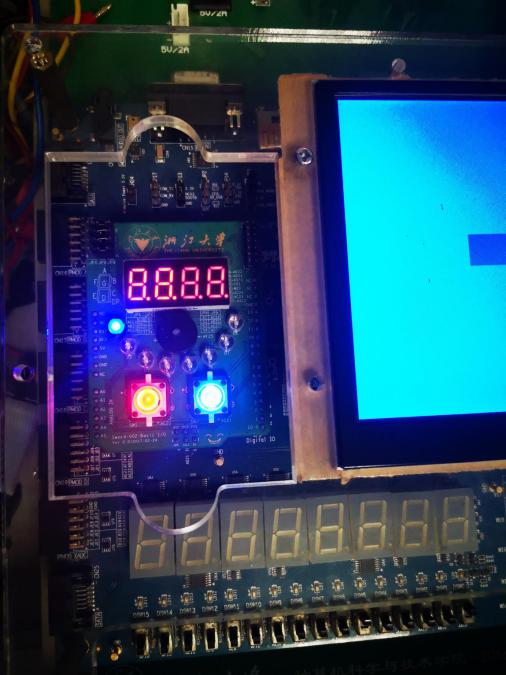


图5.3.1-2 验证结果

在实验过程中，当按下一个开关，楼道灯亮起，当按下另一个开关，楼道灯熄灭。按下第三个开关，楼道灯亮起。实现了控制楼道灯开关的功能。验证成功。

|  |  |  |  |
| --- | --- | --- | --- |
| *S*3 | *S*2 | *S*1 | *F* |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

5.3.2 真值表

# 六、讨论、心得

此次原理图较为简单，上手难度不大，收获很多。

印证了在课上学到的译码器相关知识，并利用其完成楼道中灯的控制。

同时，我对ISE平台有了更深入的了解。

**实验六——七段数码管显示译码器设计与应用**

姓名： 王祚滨 专业： 信息安全 学号： 3180104933

课程名称： 逻辑与计算机设计基础实验 同组学生姓名： 王国朝、赵卿云

指导老师： 洪奇军 实验地点： 浙江大学紫金港校区东四教学楼509实验日期：2019年 10 月 23 日

# 实验目的和要求

1. 掌握七数码管显示原理

2. 掌握七段码显示译码设计

3. 进一步熟悉Xilinx ISE 环境及SWORD实验平台

# 二、实验内容和原理

### 2.1实验内容：

任务1：原理图设计实现显示译码MyMC14495模块

任务2：用MyMC14495模块实现数码管显示

### 2.2实验原理：

2.2.1 七段数码管介绍

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。



图1 十六进制数字与七段数码管对于图

2.2.2 共阴（阳）控制

LED的正极(负极)连在一起，另一端作为点亮的控制。

共阳：正极连在一起，负极＝0，点亮

共阴：负极连在一起，正极＝1，点亮

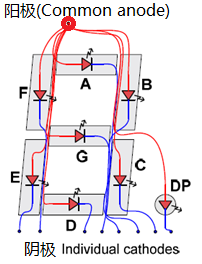


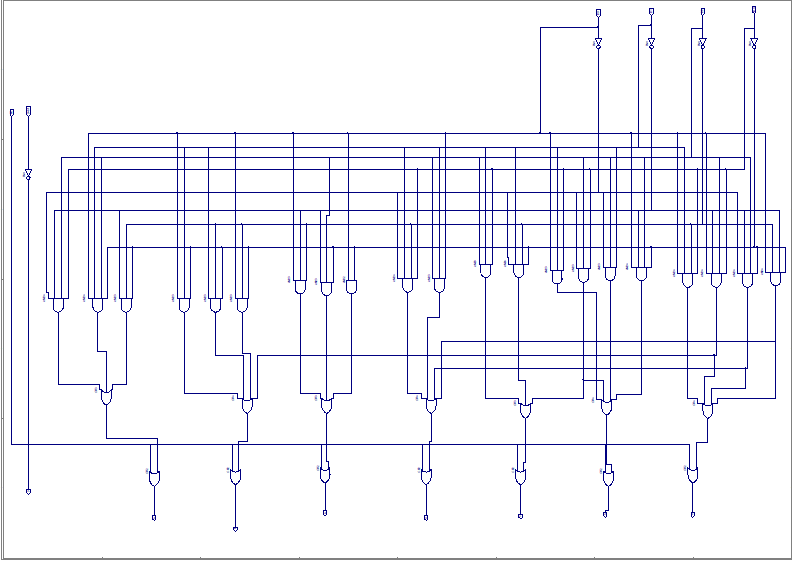
图2 七段数码管的共阳控制图

2.2.3 真值表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Hex** | **D3D2D1D0** | **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 |

图3 七段数码管与十六进制数字的对应真值表

2.2.4 运用译码器将四位的二进制数字与七段数码管静态显示对应原理图



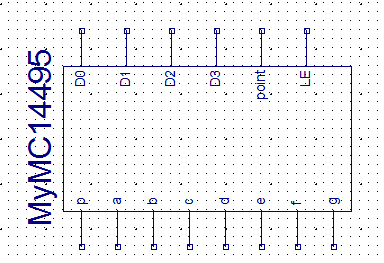


图4 MC1495原理图

# 三、主要仪器

1. 装有Xilinx ISE 14.7的计算机 1台

2. SWORD开发板 1套

# 四、操作方法与实验步骤

4.1设计实现MY\_MC14495

1. 新建工程，工程名称用MyMC14495。

2. 新建源文件，文件名称用MyMC14495。

3. 原理图方式进行设计。

4. Check Design Rules，检查错误

5. View HDL Functional Model，查看并学习Verilog HDL代码

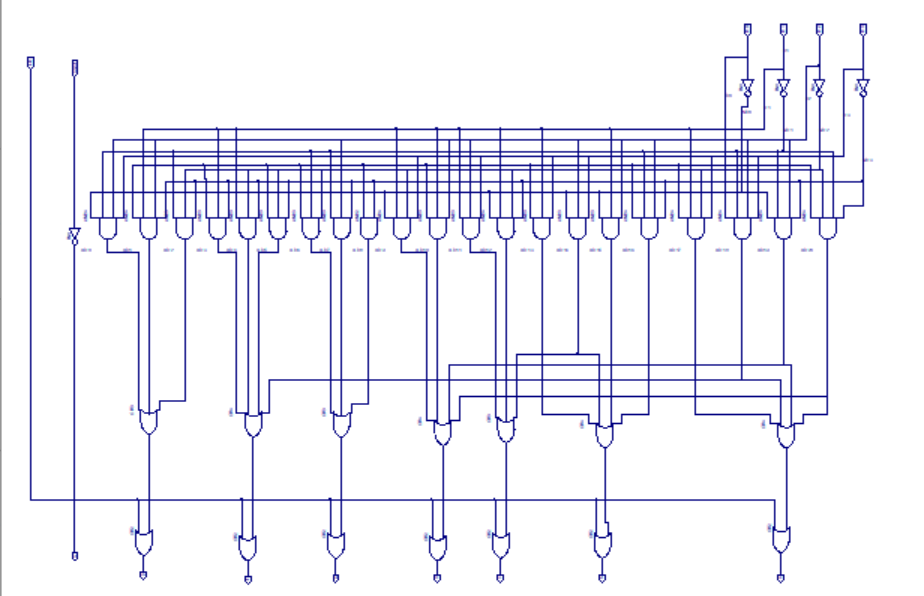
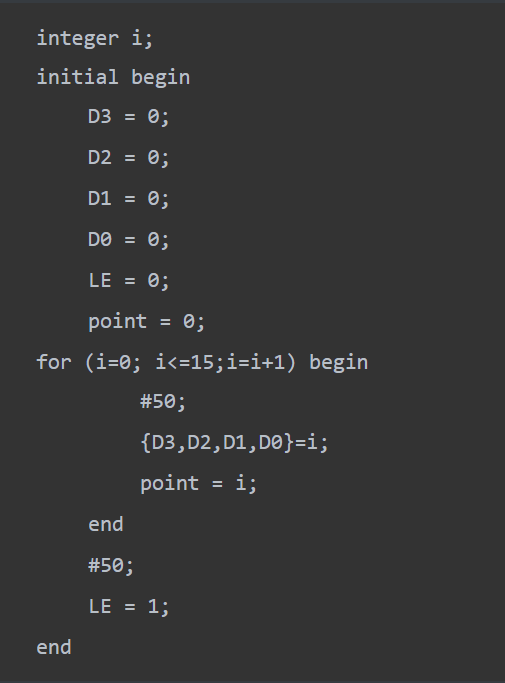


图4.1-1 静态显示7段数码管原理图

6.仿真

加入激励代码：



4.2 实现数码管显示

1. 新建工程DispNumber\_sch

2. 新建schematic文件DispNumber\_sch

3. 复制MyMC14495.sym和.vf到工程根目录，在symbols框里的第一个元件，就是MyMC14495。

4. 根据原理图设计动态显示的七段数码管

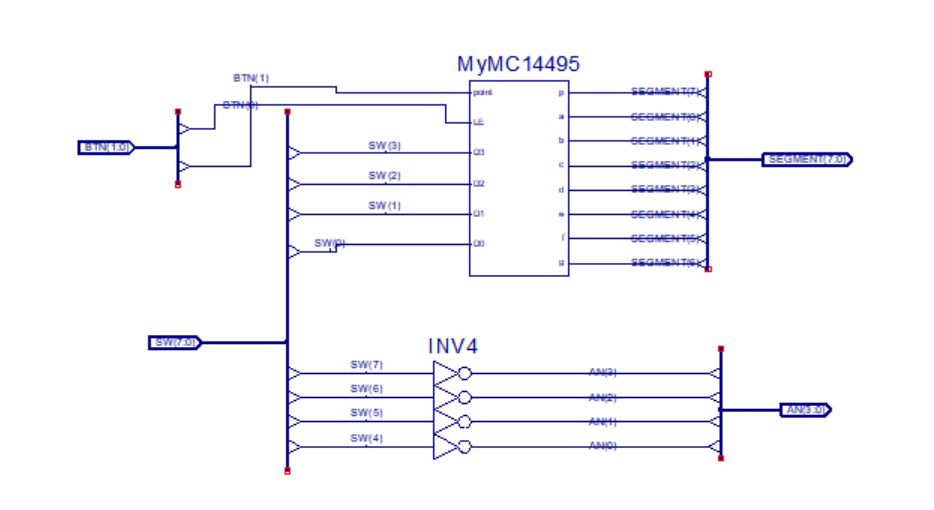


图4.2-1 动态显示的七段数码管原理图

5. UCF引脚定义

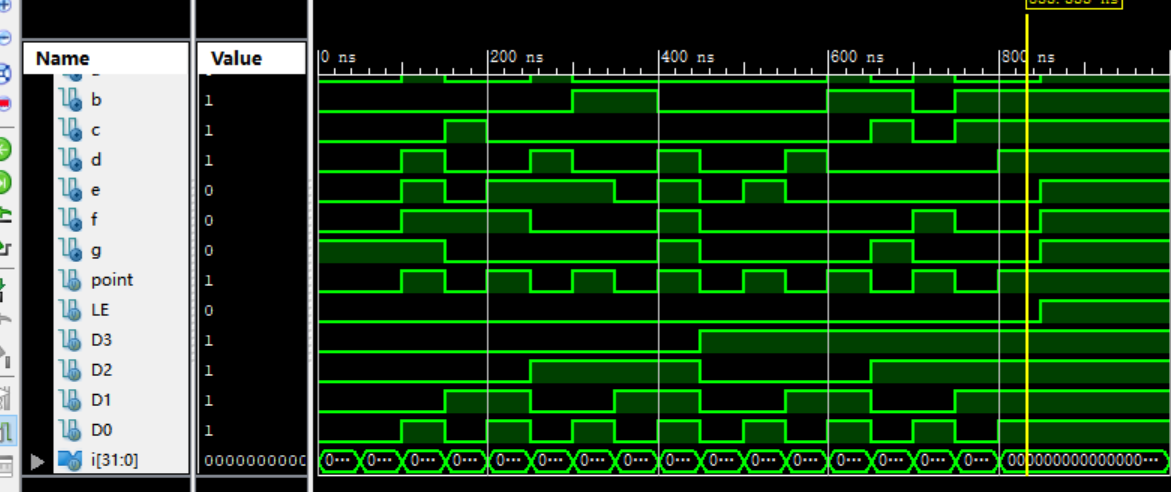
# 



# 五、实验结果与分析

5.1 原理图设计实现显示译码MyMC14495模块

5.1.1 MyMC14495仿真



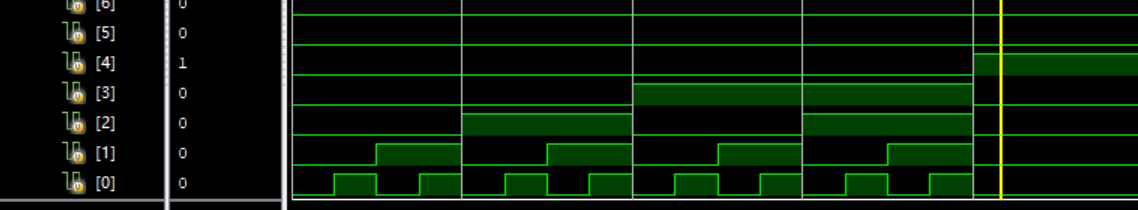


图 5.1.1-1 仿真效果图

从仿真图中可以看出，D[3:0]组成的二进制数字，会由七段数码管上的a~g显示出来，小数点也同样可以被控制是否显示。

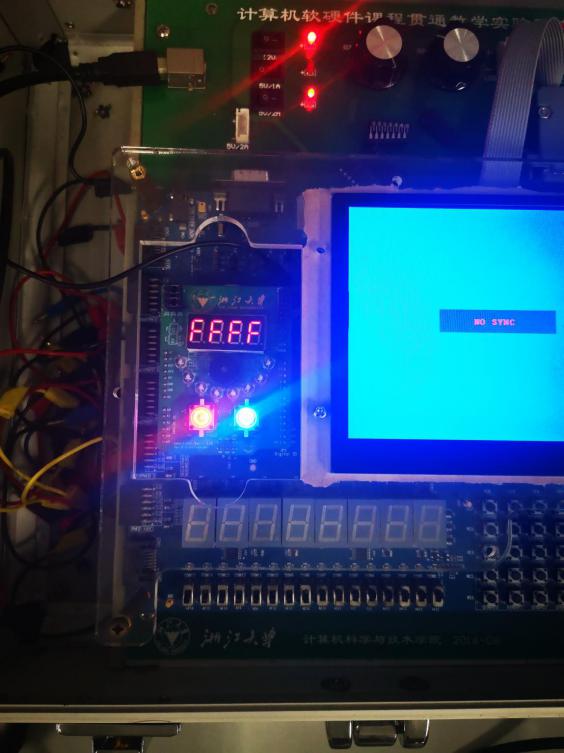
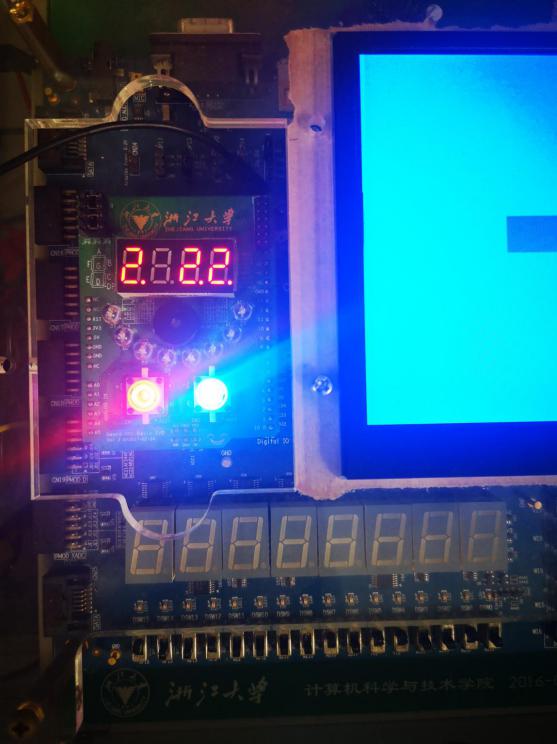
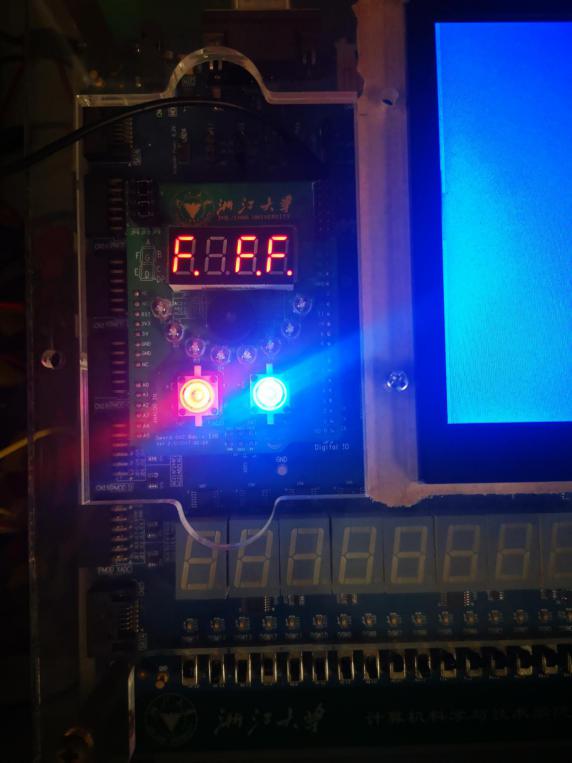
5.1.2 真值表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Hex** | **D3D2D1D0** | **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 |

经仿真验证，仿真结果符合真值表。

5.2 用MyMC14495模块实现数码管显示

5.2.1 实验结果

从实验结果可以看出，此次实验已经完成数码管的显示。

但不足之处也很明显，四个七段数码管只能显示同一位数字，而AN则控制某个七段数码管是否显示，如果要完成计分效果，需要动态显示，此操作下次实验中会进行介绍。

# 讨论、心得

在做试验前，将原理图画好，此次实验原理图很复杂，消耗了很多时间

按照课上老师所讲的以及ppt中的步骤，整体虽然花费时间很多，但还是成功将实验做了出来，不足之处就是用图表示很容易乱掉，逻辑程度不高。

**实验七——多路选择器设计与应用**

姓名： 王祚滨 专业： 信息安全 学号： 3180104933

课程名称： 逻辑与计算机设计基础实验 同组学生姓名： 王国朝、赵卿云

指导老师： 洪奇军 实验地点： 浙江大学紫金港校区东四教学楼509实验日期：2019年 10 月 30 日

# 一、实验目的和要求

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

**二、实验内容和原理**

实验内容：

任务1：数据选择器设计

任务2：记分板设计

实验原理：

1. 4选一多路选择器：MUX4to1

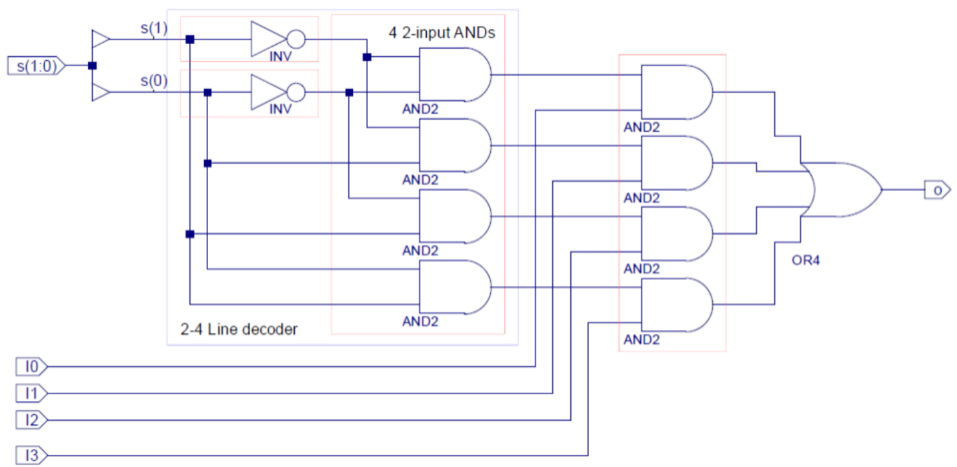
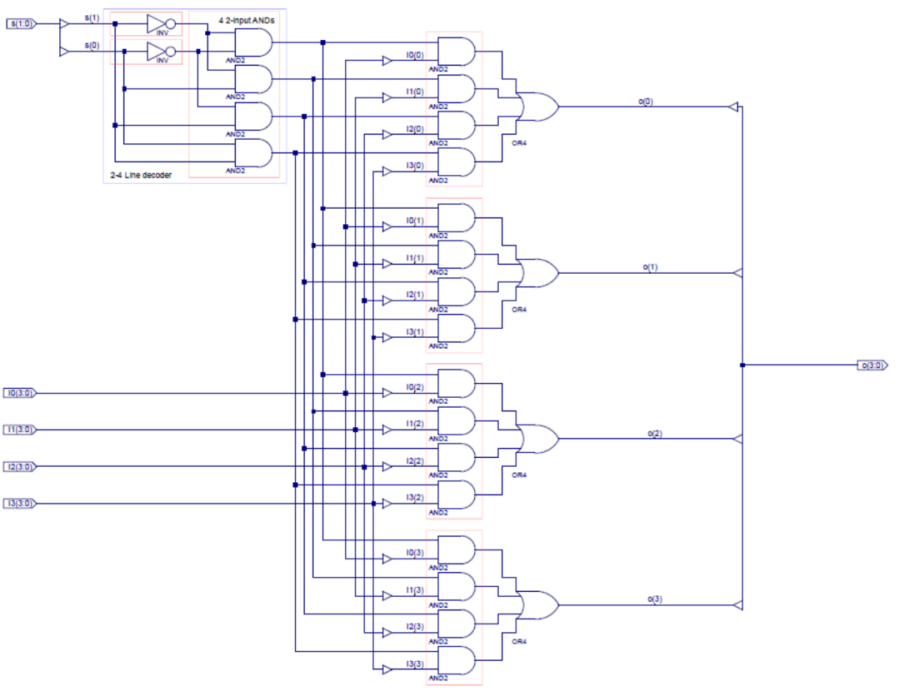


图1 4选1多路选择器原理图



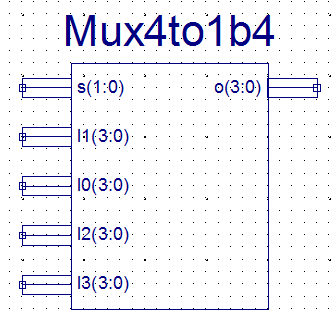


图2 4位四选一扩展（MUX4to1b4）

2. 动态扫描显示



图3 动态扫描显示方案

3. 多路选择器应用：4位七段显示扫描控制

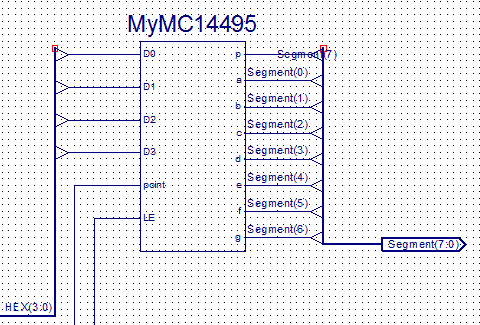
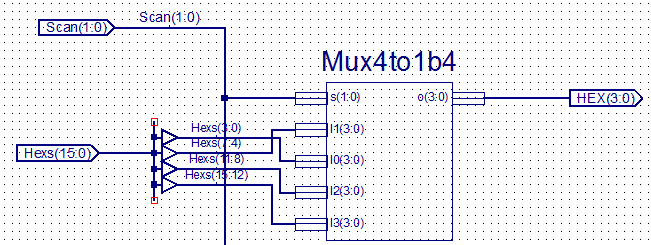


图4 数字显示



图5 小数点显示及使能信号

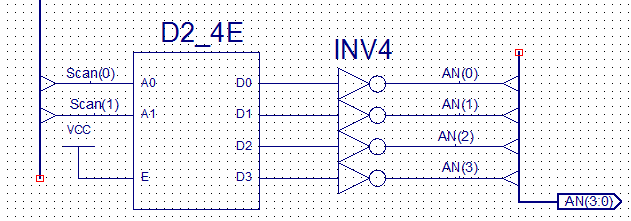
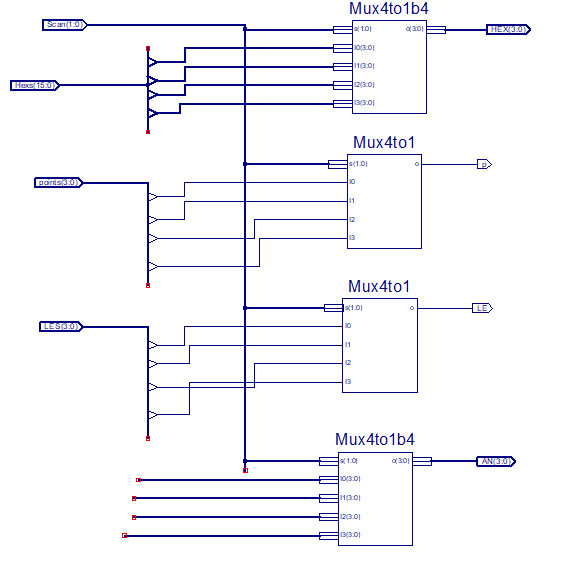
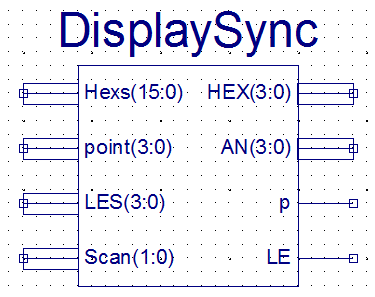


图6 选择一个七段数码管显示

4. DisplaySync.sch原理图





5 时钟计数分频器（代码）

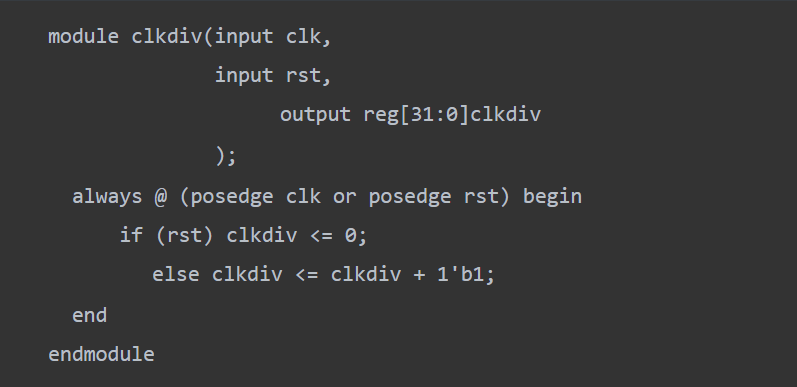


图 7 clkdiv图

6. 设计disp\_num显示模块

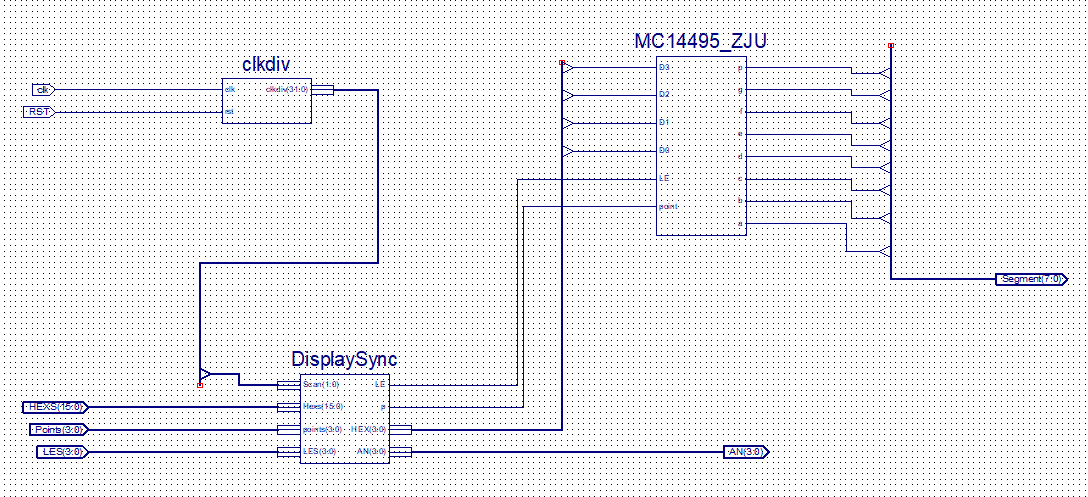
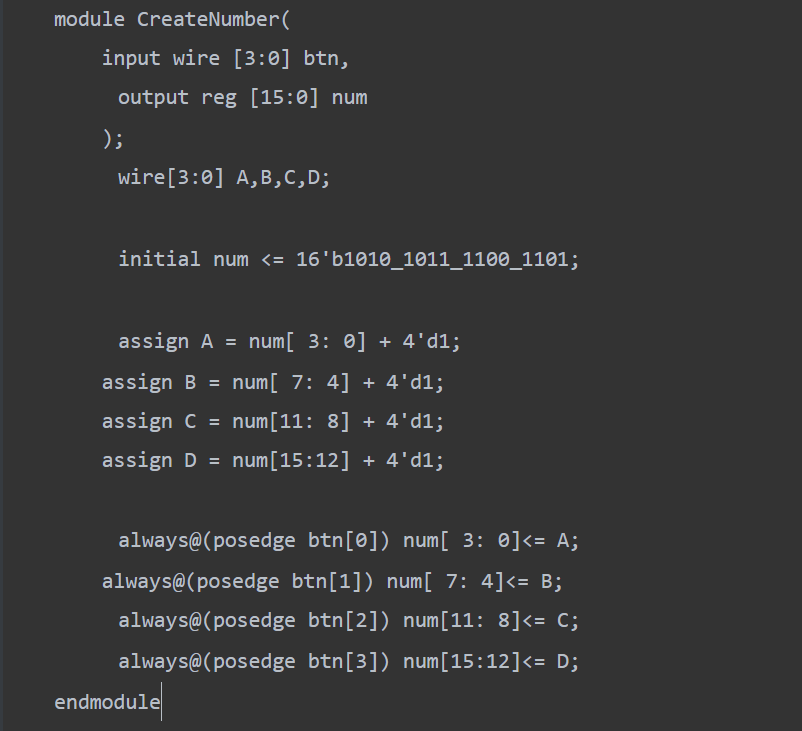


图8 disp\_num原理图

7. CreateNumber模块



# 三、主要仪器设备

# 1. 装有Xilinx ISE 14.7的计算机 1台

# 2. SWORD开发板

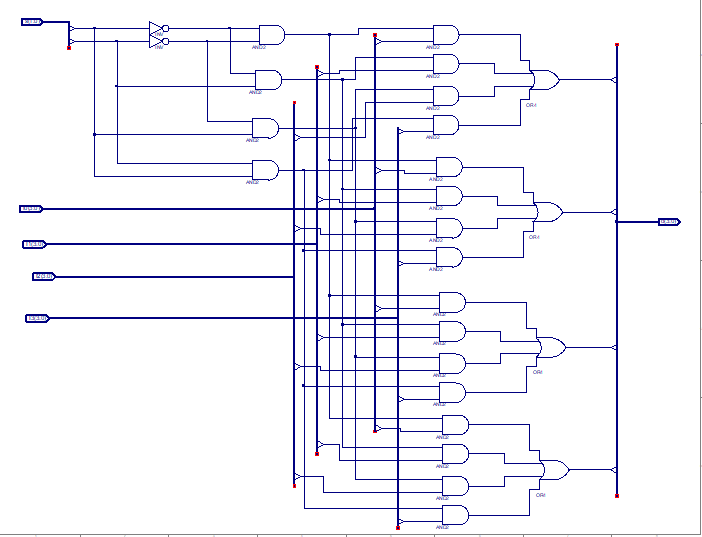
# 四、操作方法与实验步骤

**任务1：数据选择器设计**

1. 新建工程，工程名称用Mux4to1b4\_sch。

2. 新建源文件，类型是Schematic，文件名称用Mux4to14b。

3. 原理图方式进行设计



**任务2：记分板设计**

1. 新建工程，工程名称用ScoreBoard，Top Level Source Type用HDL

2. 根据原理设计动态扫描同步输出模块

3. 根据原理设计通用计数分频模块

4. 新建源文件，类型是Schematic，文件名称用disp\_num

5. 原理图方式进行设计显示模块

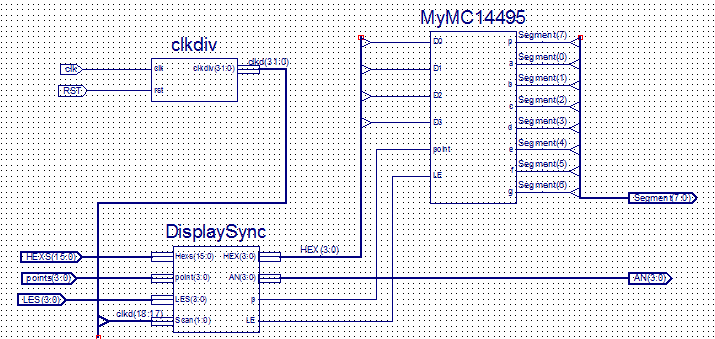
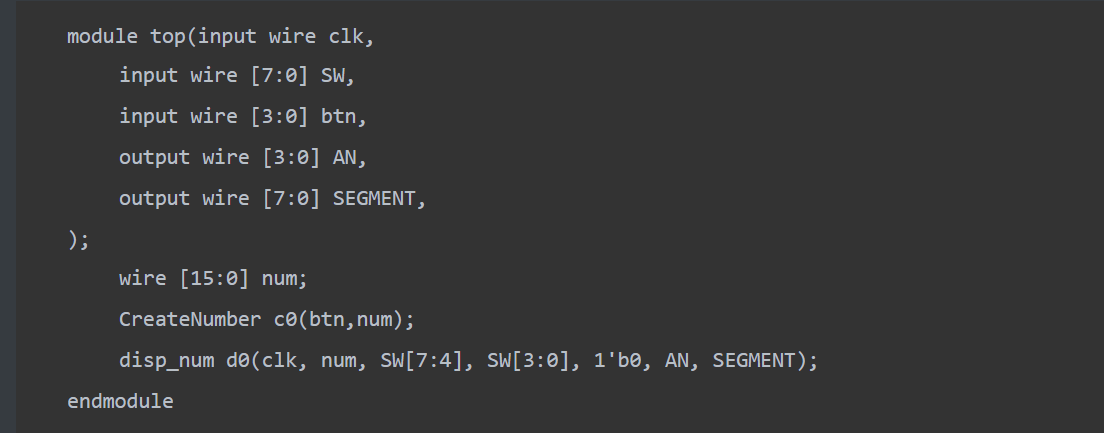
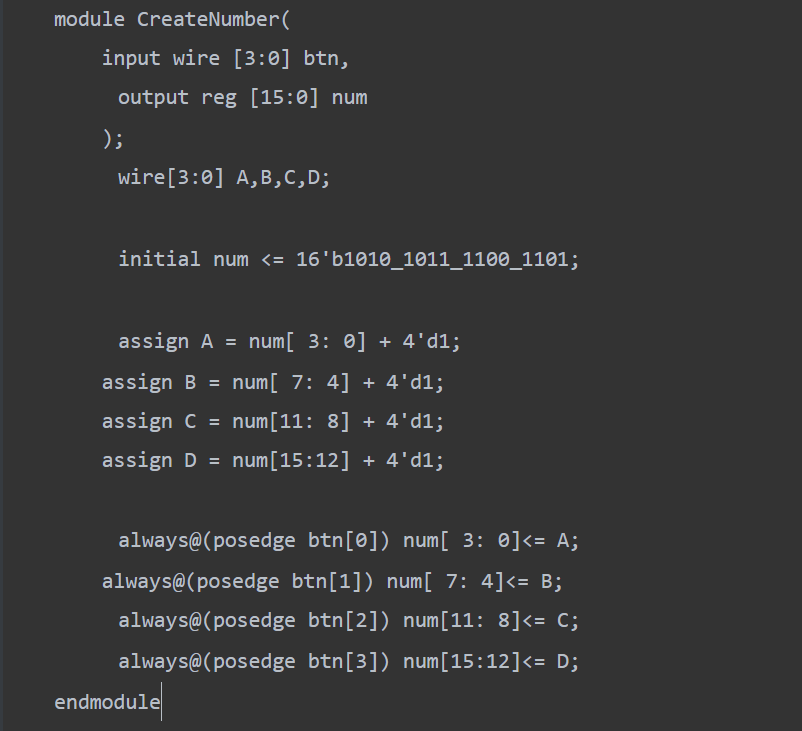
****

图4.2.5-1 动态显示计分板

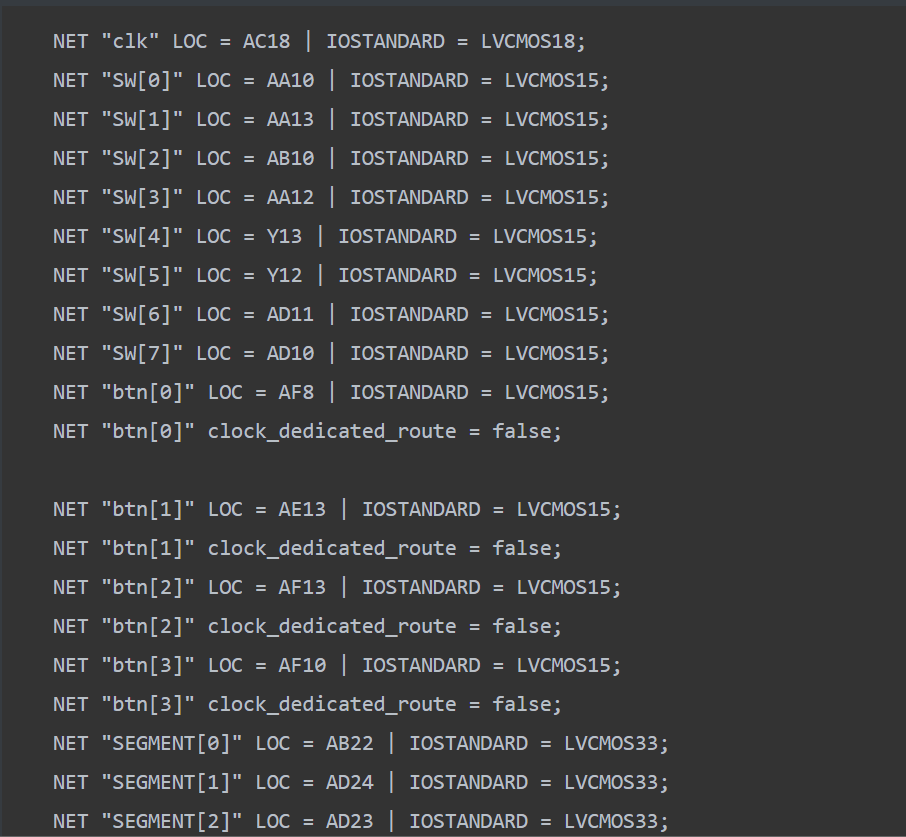
6. 新建源文件top，并右键设为“Top Module”



7. CreateNumber代码



8. UCF引脚定义





# 五、实验结果与分析

**5.1数据选择器设计**

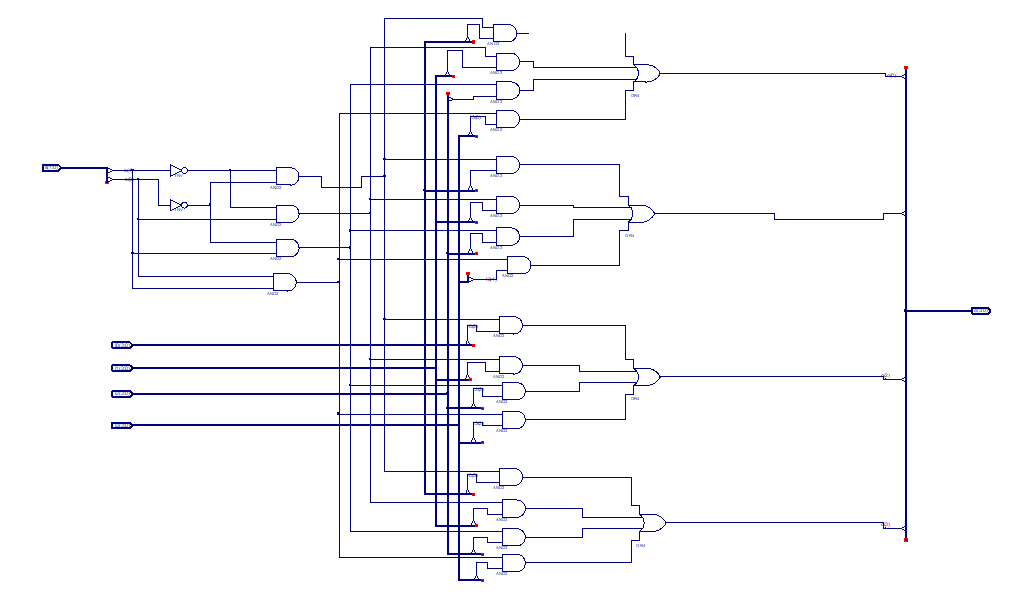


图 1 Mux4to1b4 原理图

5.2 记分板应用设计

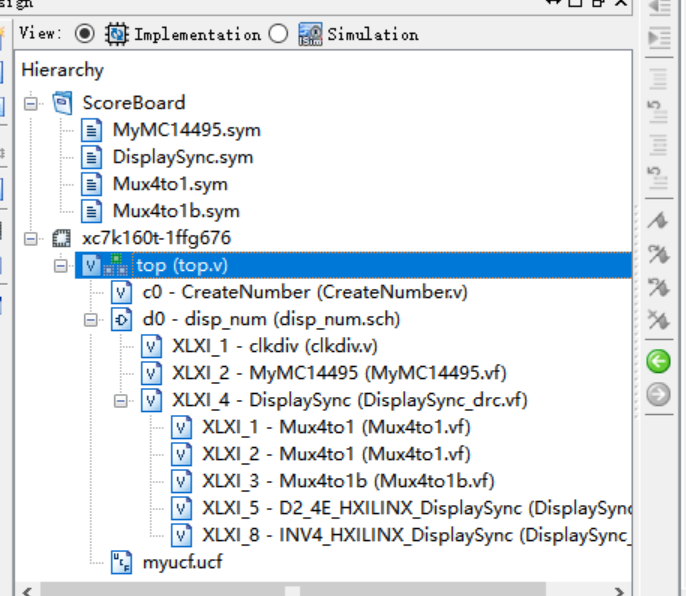


图1：ScoreBoard工程目录

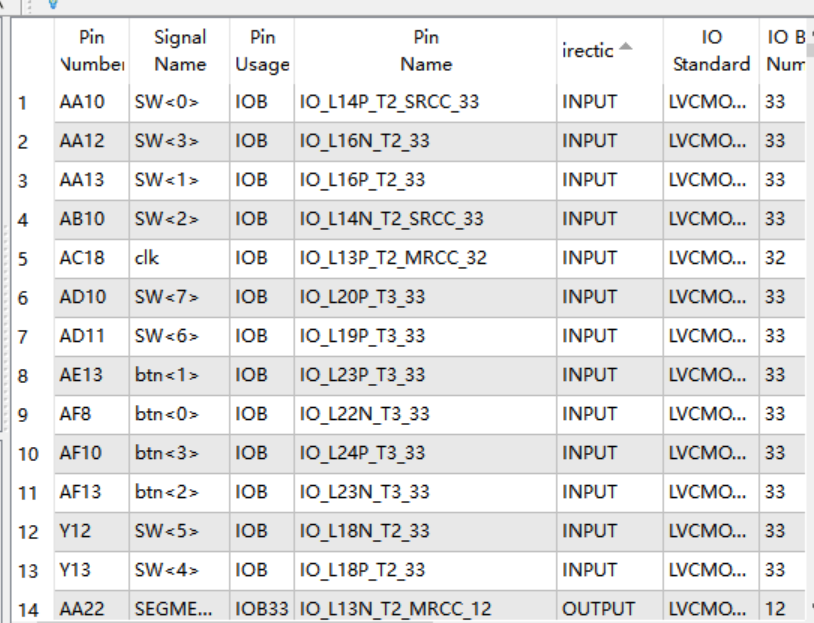


图2：pinout report

5.2.1实验结果



最左端四个开关表示控制数字加一，最右边四位表示控制小数点有无，左边四位表示控制数字显示，一个七段数码管只显示一个数字，但通过动态扫描和人眼视觉残留，给人一种同时显示四个数字的错觉。

同时，在实验中发现一个问题，当我把开关置于中间位置时（既不开也不关），对应位置的数码管会开始乱闪，咨询过老师后发现是“去抖动”没有做好，下次实验要学习此内容，我们认为开关拨了一次但机器认为拨动了很多次，所以会开始乱闪。

# IMG_20191030_184802IMG_20191030_184743IMG_20191030_184732

在图二中，最右边四位控制小数点有无的开关，前三个处于高电平状态，与出现的三个小数点吻合，而中间四位控制数字有无的开关，中间两位处于高电平状态，因此数字不显示。

# 六、讨论、心得

本次实验中，独立建造了四个组件，工程量较大，但由于之前有的建立sch文件的经验，操作起来并不困难，但一定要在每一个组件做完后仔细检查，否则做到最后发现组件存在问题改起来很困难，这次实验我便是在其中一个组件出了问题，调试了近1个小时才完成。

同时，本实验使用top-down设计思路，将任务逐渐简化成各个组件，使整个过程思路清晰，而且各个组件形成了可复用组件，比如此次实验就使用了实验六制作的组件。

附：

**个人照片：**

****