

# 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: 刘晨

学 院: 计算机科学与技术学院

系:

专 业: 图灵班

学 号: 3190104666

指导教师: 董亚波

2020 年 10 月 29 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: 7 段数码管显示译码器设计与应用

学生姓名: 刘晨 专业: 图灵 1901 学号: 3190104666

同组学生姓名: 林初涵 指导老师: 董亚波

实验地点: 东 4-509 实验日期: 2020 年 10 月 29 日

## 一、实验目的和要求

掌握七数码管显示原理

掌握七段码显示译码设计

进一步熟悉 Xilinx ISE 环境及 SWORD 实验平台

## 二、实验内容和原理

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

原理:

7 数码管是一个由 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。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 则点亮

共阴: 负极连在一起, 正极=1 则点亮

**多位七段数码管显示原理:**

静态显示:

每个 7 段码对应一个显示译码电路

动态扫描显示: 时分复用显示:

利用人眼视觉残留, 一个 7 段码译码电路分时为每个 7 段码提供译码

控制时序：

用定时计数信号控制公共极，分时输出对应七段码的显示信号，采用动态扫描的方式

4 位七段码结构：

正极为公共端，七段信号并联

分时控制示意：

动态扫描

低电平与输入显示对应

共阳：低电平控制

分时送 a~g, p

可用序列信号控制

### 三、实验过程和数据记录

#### 任务 1：原理图设计实现显示译码 MyMC14495 模块

##### 1.设计实现 MY\_MC14495

新建工程，工程名称用 MyMC14495；

新建源文件，文件名称用 MyMC14495；

按照原理图方式进行设计；

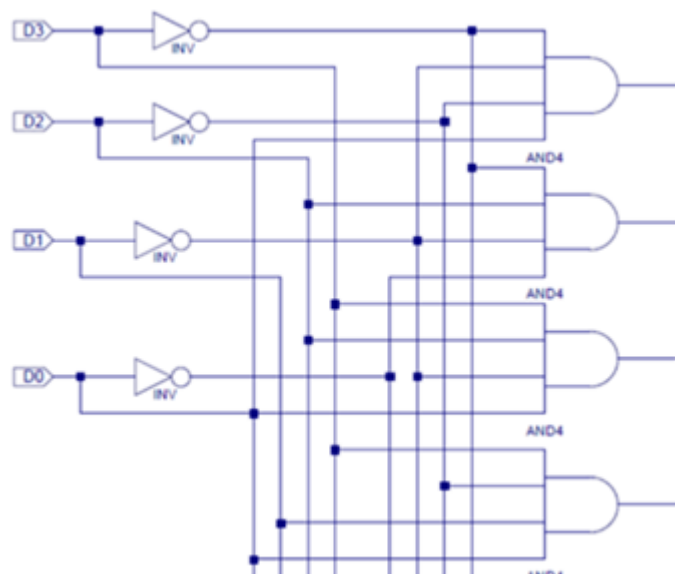


图 1 原理图（部分）

Check Design Rules, 检查错误

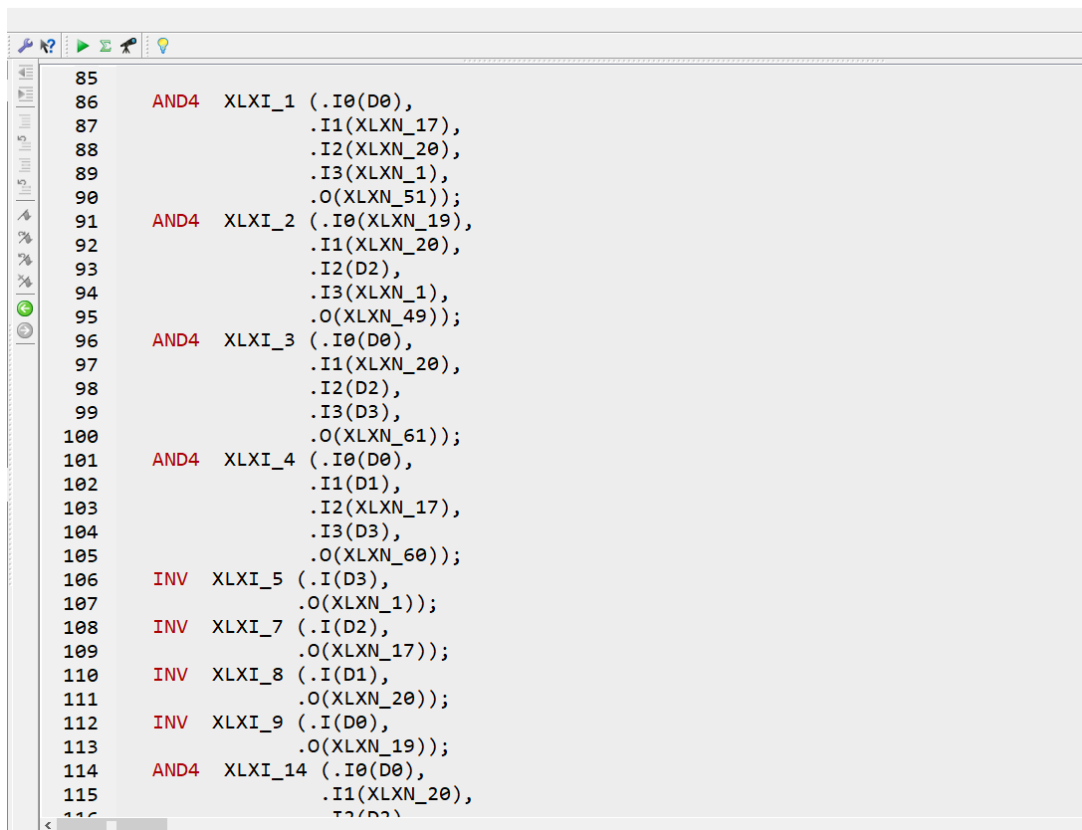


图 2 硬件描述代码

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

## 2.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;
    LE = 0;
    point = 0;
    for (i=0; i<=15;i=i+1) begin
        #50;
        {D3,D2,D1,D0}=i;
        point = i;
    end

    #50;

```

```
LE = 1;  
end
```

得到的仿真信号如下：

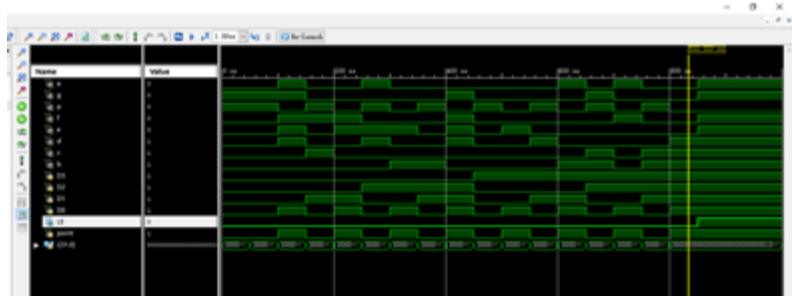


图 3 仿真信号图示

### 3.生成逻辑符号图

Create Schematic Symbol，系统生成 MyMC14495 模块的逻辑符号图文件，文件后缀. sym，符号图位于工程根目录

自动生成的符号可修改，修改方法如下：可以用 Tools 菜单的 Symbol Wizard，也可以打开. sym 文件直接修改

使用时必须复制. sym 和. sch 到对应工程目录

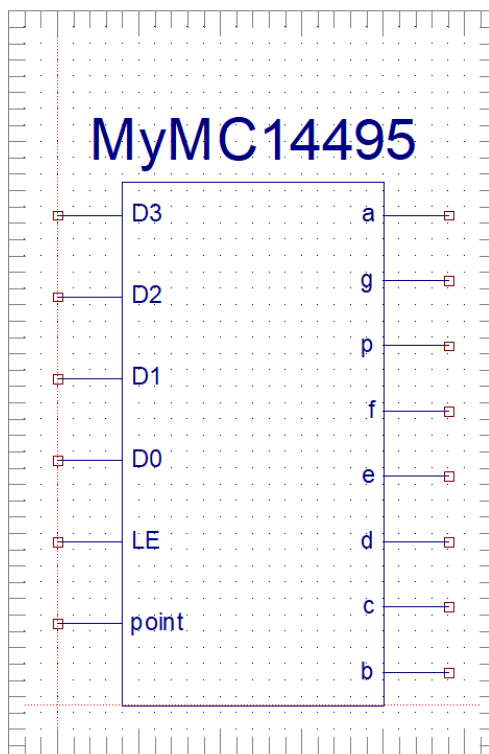


图 4 逻辑符号图图示

## 任务 2：实现数码管显示

## 1. 调用 MyMC14495

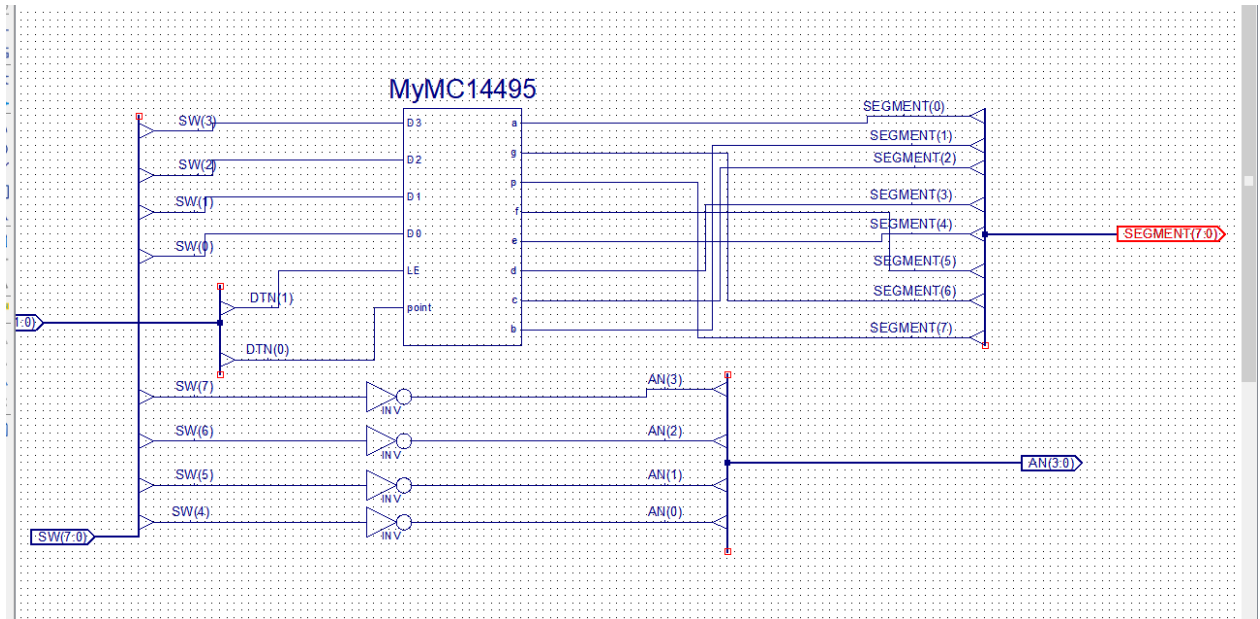


图 5 调用 MyMC14495 的图示

## 2. 下载验证

## 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

 $\tilde{a}_g, p$ 

ucf 文件如下:

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

3.生成 bit 文件并下载到开发板

如图是开发板导入 bit 文件后的初始状态. 调试结果如下:

(1) 输出数字 “0”

开关	SW[3]	SW[2]	SW[1]	SW[0]	SW[7]	SW[6]	SW[5]	SW[4]	SW[14]	SW[15]
状态	0	0	0	0	0	1	0	0	0	0

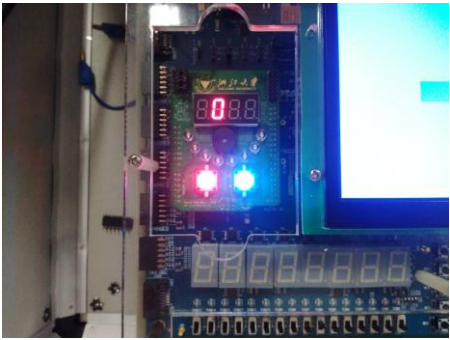


图 6 数字 0

(2) 输出数字 “1”

开关	SW[3]	SW[2]	SW[1]	SW[0]	SW[7]	SW[6]	SW[5]	SW[4]	SW[14]	SW[15]
状态	0	0	0	1	0	1	0	0	0	0



图 7 数字 1

(3) 输出数字“4”

开关	SW[3]	SW[2]	SW[1]	SW[0]	SW[7]	SW[6]	SW[5]	SW[4]	SW[14]	SW[15]
状态	0	1	0	0	0	1	0	0	0	0



图 8 数字 4

(4) 输出数字“8” “9”

开关	SW[3]	SW[2]	SW[1]	SW[0]	SW[7]	SW[6]	SW[5]	SW[4]	SW[14]	SW[15]
数字 8	1	0	0	0	0	1	0	0	0	0
数字 9	1	0	0	1	0	1	0	0	0	0

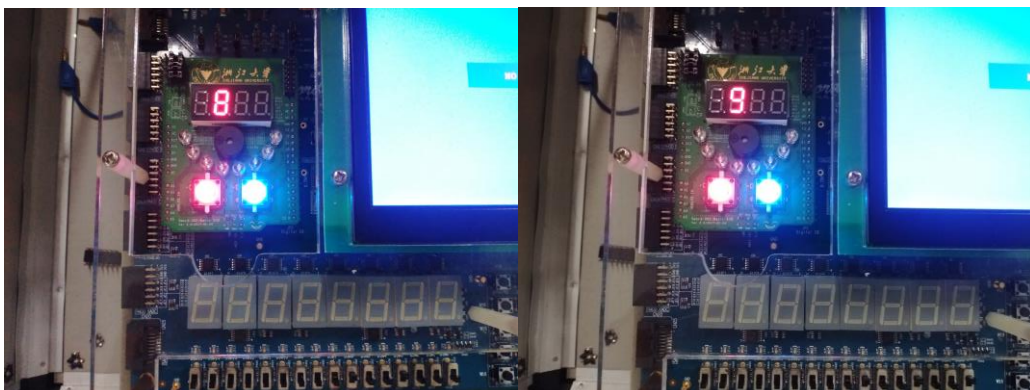




图9 数字“8”与“9”

(5) 输出字母 “A” “C” “D” “E”

开关	SW[3]	SW[2]	SW[1]	SW[0]	SW[7]	SW[6]	SW[5]	SW[4]	SW[14]	SW[15]
字母 A	1	0	1	0	0	1	0	0	0	0
字母 C	1	1	0	0	0	1	0	0	0	0
字母 D	1	1	0	1	0	1	0	0	0	0
字母 E	1	1	1	0	0	1	0	0	0	0

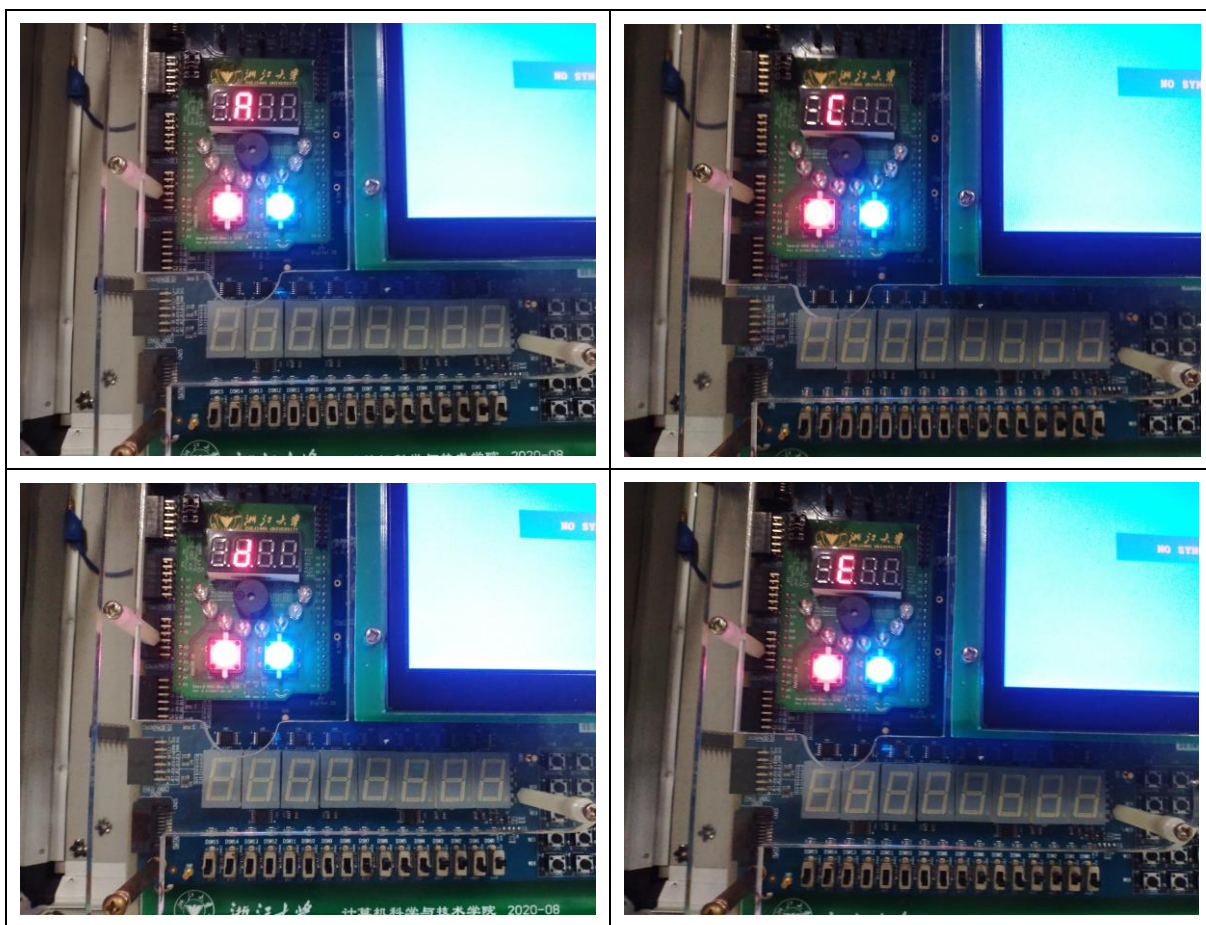


图 10 字母“A” “C” “D” “E”

(6) 输出 “F.F.F.F.”

[illegible]

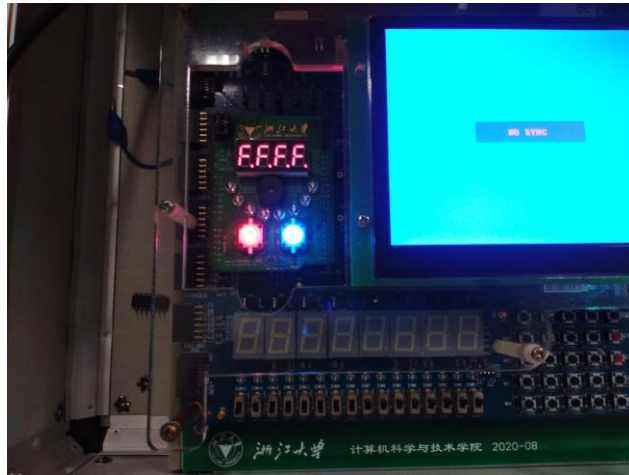


图 11 F.F.F.F

#### 四、实验结果分析

##### ① 分析生成的硬件描述代码

硬件描述代码符合预期，并能输出正确结果。

##### ② 分析仿真结果

结合仿真代码，仿真给的激励是依次改变 D3，D2，D1，D0 的值，在这种激励下得到输出的仿真结果是不同的数码管的仿真结果，这个结果符合预期。这样的结果在开发板上体现如下表。

##### ③ 分析开发板结果

开发板结果符合设计要求。

开发板调试结果汇总：

输出结果	D3	D2	D1	D0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0

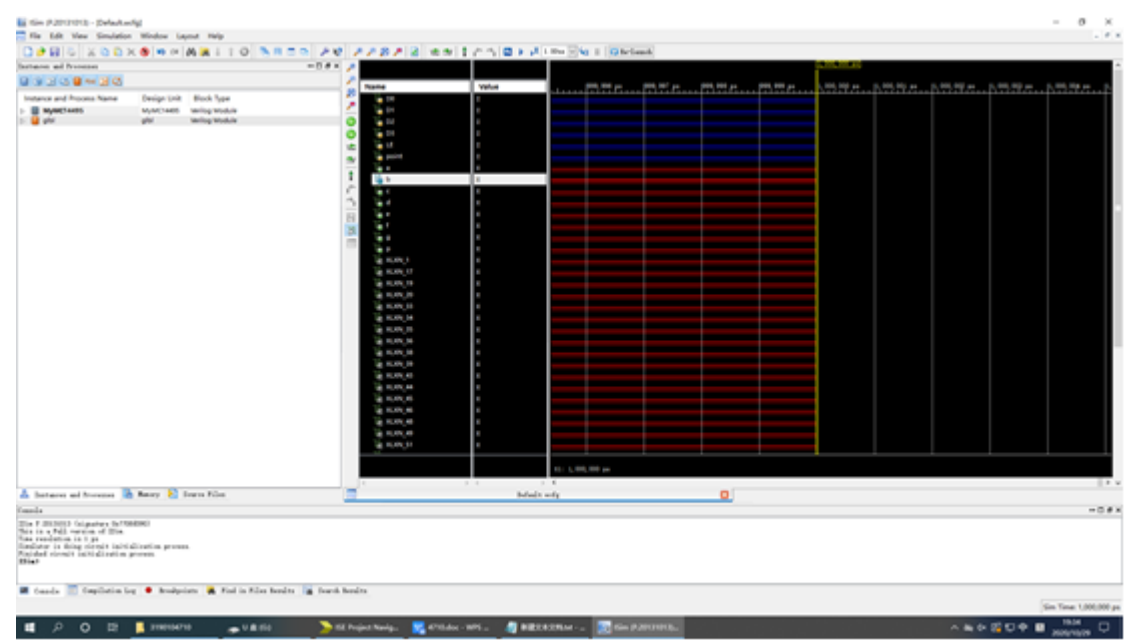
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

开关 4 5 6 7 分别控制四个数字是否亮；

开关 14 控制是否显示数字；

开关 15 控制是否显示小数点。

### 五、讨论与心得



在做实验当中，第一次的仿真结果没有任何输出，所有的输入值都是 X。于是我检查了整个系统的电路，发现并没有出现什么问题；而我最后在讨论中，得知由于我并没有将仿真的文件插入到建立的 project 当中，所以根本没有检测到仿真的那个文件，因故没有输出结果了。