

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： 龙永奇

学 院： 计算机科学与技术学院

系： 本系

专 业： 计算机科学与技术

学 号： 3220105907

指导教师： 董亚波

2023 年 11 月 3 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：____ 龙永奇 ____ 专业：____ 计算机科学与技术 ____ 学号：____ 3220105907 ____

同组学生姓名：____ 贾一多 ____ 指导老师：____ 董亚波 ____

实验地点：____ 东 4-509 ____ 实验日期：____ 2023 年 10 月 26 日 ____

一、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

内容：

1. 原理图设计实现显示译码 MyMC14495 模块
2. 用 MyMC14495 模块实现数码管显示

原理：

1. 显示译码 MyMC14495 模块设计

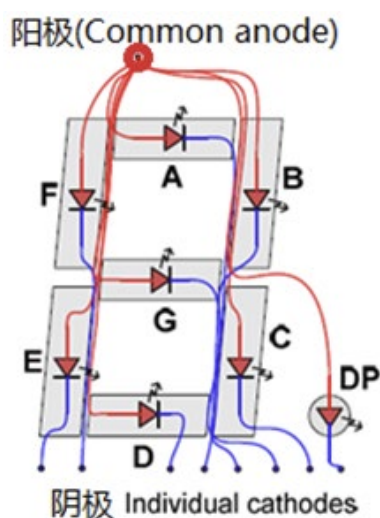
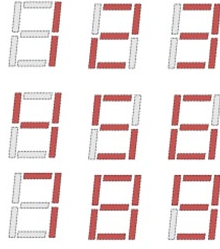


图 2.1 七段数码管

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

X	a	b	c	d	e	f	g
0	0	0	0	0	0	1	
1	1	0	0	1	1	1	
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0
A	0	0	0	1	0	0	0
B	1	1	0	0	0	0	0
C	0	1	1	0	0	0	1
D	1	0	0	0	0	1	0
E	0	1	1	0	0	0	0
F	0	1	1	1	0	0	0



0 = on 1 = off

图 2.2 1-9 数字对应真值表

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。

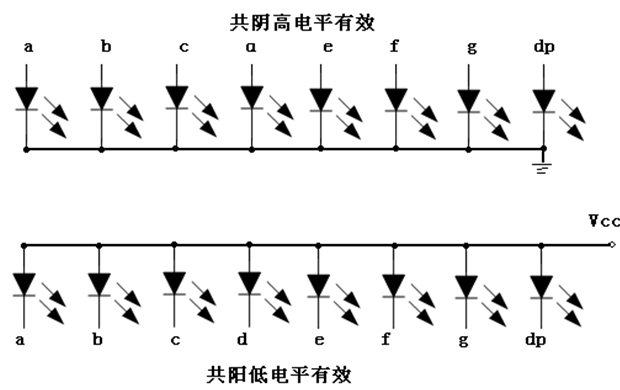


图 2.3 共阳、共阴控制

由于输出逻辑 0 的电路负载能力更强，因此常用共阳输出
真值表如下：

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

图 2.4 真值表

由卡诺图化简可得：

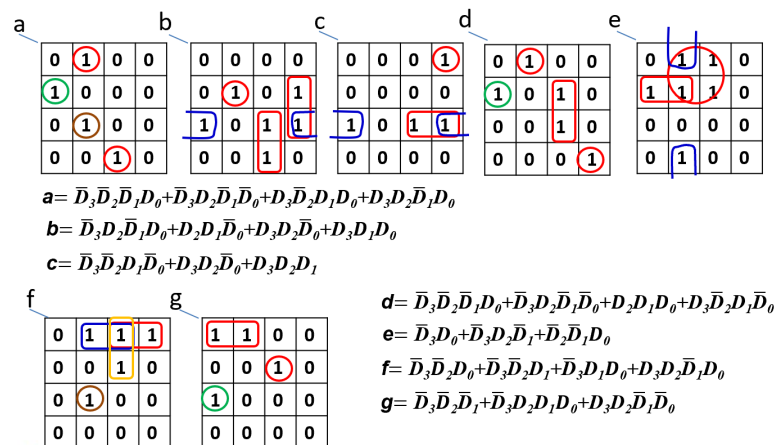


图 2.5 化简结果

根据化简结果可绘制出电路图（省略 VCR(Pin11)和 h+i (Pin4) 功能）：

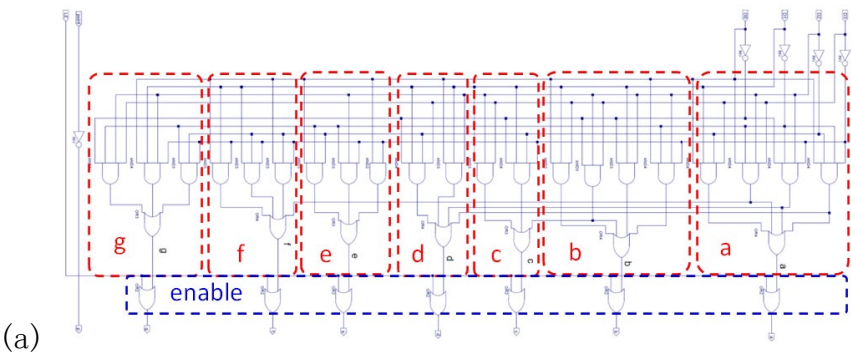


图 2.6 电路图

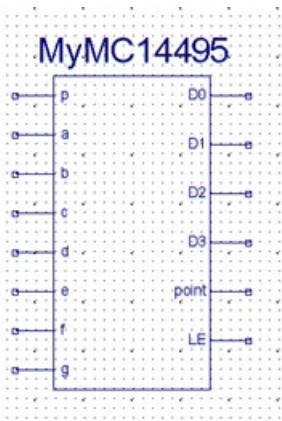


图 2.7 符号图

2. 多位七段数码管显示原理

(a) 静态显示

每个 7 段码对应一个显示译码电路

(b) 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

(c) 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：动态 扫描

(d) 4 位七段码结构

正极：公共端

七段信号并联

3. 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

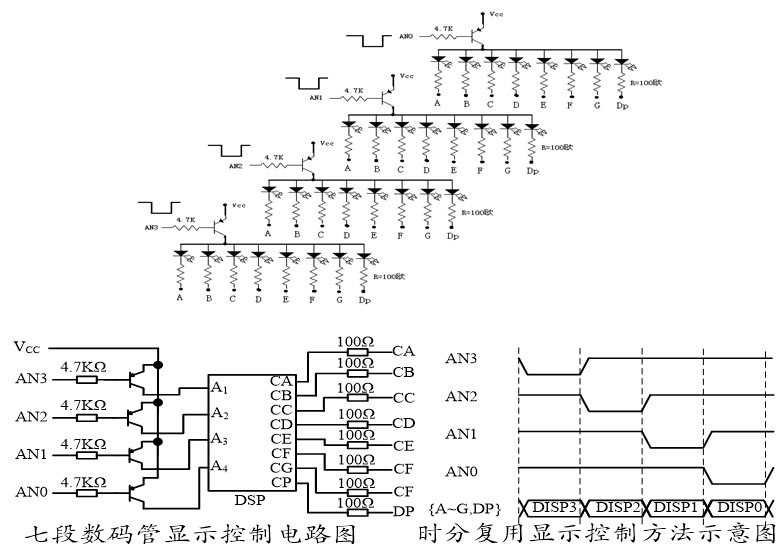


图 2.8 分段控制示意图

三、实验过程和数据记录

3.1 原理图设计实现显示译码 MyMC14495 模块

1. 设计实现 MY_MC14495

(a) 在 ISE 中点击 File 选项卡，点击 New Project，工程名为

MyMC14495

(b) 在 Sources 窗口中右键选择 New Sources 新建源文件向导中选择源

文件类型为 Schematic, 输入文件名 MyMC14495, 勾选 Add to Project
(c) 使用 Symbols 和 Schematic Editor 输入原理图

由于本次设计电路图较大, 可用 ISE 的 Edit->Change Sheet Size 菜单项来改变原理图绘图区的尺寸:

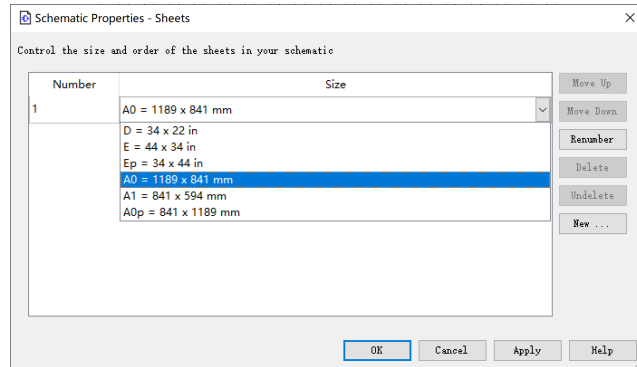
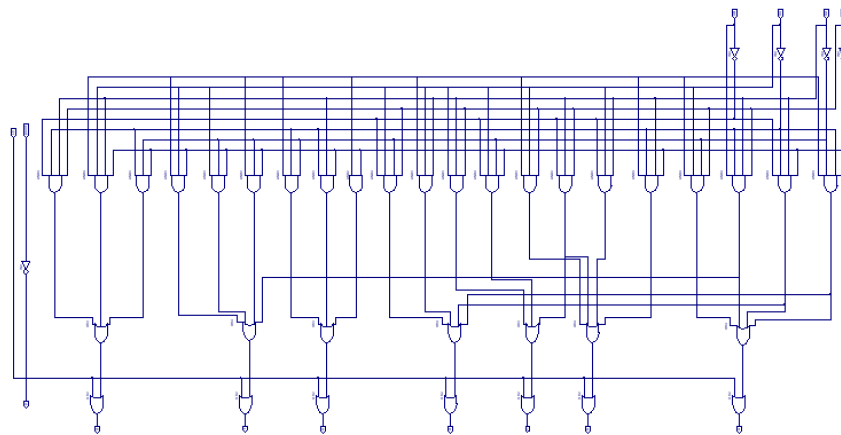


图 3.1 改变设计图尺寸



3.2 电路图

- (d) 检查错误并查看电路的硬件描述代码
- (e) 在 Sources 窗口中选择 Sources for: Synthesis/Implementation, 选中 MyMC14495 图标, 在 Processes 窗口 Processes 选项卡中展开 Design Utilities 双击 Check Design Rules 无误后双击 View HDL Functional Model, 如下图:


```
output g;
output p;

wire XLXN_1;
wire XLXN_2;
wire XLXN_4;
wire XLXN_5;
wire XLXN_9;
wire XLXN_10;
wire XLXN_11;
wire XLXN_12;
wire XLXN_14;
wire XLXN_16;
wire XLXN_17;
wire XLXN_21;
wire XLXN_22;
wire XLXN_24;
wire XLXN_25;
wire XLXN_26;
wire XLXN_27;
wire XLXN_28;
wire XLXN_30;
wire XLXN_35;
wire XLXN_38;
wire XLXN_47;
wire XLXN_48;
wire XLXN_49;
wire XLXN_50;
wire XLXN_51;
wire XLXN_52;
wire XLXN_53;
wire XLXN_63;
wire XLXN_64;
wire XLXN_65;
wire XLXN_68;

AND4 XLXI_1 (.I0(XLXN_65),
              .I1(XLXN_64),
              .I2(D2),
              .I3(D3),
              .O(XLXN_1));
AND4 XLXI_2 (.I0(D0),
              .I1(D1),
              .I2(D2),
```



```

        .I3(XLXN_68),
        .O(XLXN_2));
AND3  XLXI_81 (.I0(XLXN_64),
               .I1(XLXN_63),
               .I2(XLXN_68),
               .O(XLXN_4));
AND3  XLXI_82 (.I0(D0),
               .I1(D1),
               .I2(XLXN_68),
               .O(XLXN_5));
AND3  XLXI_83 (.I0(D1),
               .I1(XLXN_63),
               .I2(XLXN_68),
               .O(XLXN_9));
AND3  XLXI_84 (.I0(D0),
               .I1(XLXN_63),
               .I2(XLXN_68),
               .O(XLXN_10));
AND3  XLXI_85 (.I0(D0),
               .I1(XLXN_64),
               .I2(XLXN_63),
               .O(XLXN_11));
AND3  XLXI_86 (.I0(XLXN_64),
               .I1(D2),
               .I2(XLXN_68),
               .O(XLXN_12));
AND2  XLXI_95 (.I0(D0),
               .I1(XLXN_68),
               .O(XLXN_14));
AND4  XLXI_96 (.I0(XLXN_65),
               .I1(D1),
               .I2(XLXN_63),
               .I3(D3),
               .O(XLXN_16));
AND3  XLXI_99 (.I0(D0),
               .I1(D1),
               .I2(D2),
               .O(XLXN_17));
AND3  XLXI_100 (.I0(D1),
                .I1(D2),
                .I2(D3),
                .O(XLXN_21));
AND4  XLXI_101 (.I0(XLXN_65),
                .I1(D1),

```

```

        .I2(XLXN_63),
        .I3(XLXN_68),
        .O(XLXN_22));
AND3  XLXI_102 (.I0(D0),
        .I1(D1),
        .I2(D3),
        .O(XLXN_24));
AND3  XLXI_103 (.I0(XLXN_65),
        .I1(D2),
        .I2(D3),
        .O(XLXN_25));
AND3  XLXI_104 (.I0(XLXN_65),
        .I1(D1),
        .I2(D2),
        .O(XLXN_26));
AND4  XLXI_108 (.I0(D0),
        .I1(XLXN_64),
        .I2(D2),
        .I3(XLXN_68),
        .O(XLXN_27));
AND4  XLXI_109 (.I0(D0),
        .I1(D1),
        .I2(XLXN_63),
        .I3(D3),
        .O(XLXN_28));
AND4  XLXI_110 (.I0(D0),
        .I1(XLXN_64),
        .I2(D2),
        .I3(D3),
        .O(XLXN_38));
AND4  XLXI_111 (.I0(XLXN_65),
        .I1(XLXN_64),
        .I2(D2),
        .I3(XLXN_68),
        .O(XLXN_30));
AND4  XLXI_112 (.I0(D0),
        .I1(XLXN_63),
        .I2(XLXN_64),
        .I3(XLXN_68),
        .O(XLXN_35));
OR3   XLXI_119 (.I0(XLXN_1),
        .I1(XLXN_2),
        .I2(XLXN_4),
        .O(XLXN_53));

```

```

OR4  XLXI_120 (.I0(XLXN_5),
              .I1(XLXN_9),
              .I2(XLXN_10),
              .I3(XLXN_38),
              .O(XLXN_52));
OR3  XLXI_121 (.I0(XLXN_11),
              .I1(XLXN_12),
              .I2(XLXN_14),
              .O(XLXN_51));
OR4  XLXI_122 (.I0(XLXN_16),
              .I1(XLXN_17),
              .I2(XLXN_30),
              .I3(XLXN_35),
              .O(XLXN_50));
OR4  XLXI_123 (.I0(XLXN_24),
              .I1(XLXN_25),
              .I2(XLXN_26),
              .I3(XLXN_27),
              .O(XLXN_48));
OR4  XLXI_124 (.I0(XLXN_28),
              .I1(XLXN_38),
              .I2(XLXN_30),
              .I3(XLXN_35),
              .O(XLXN_47));
OR3  XLXI_125 (.I0(XLXN_21),
              .I1(XLXN_22),
              .I2(XLXN_25),
              .O(XLXN_49));
INV  XLXI_126 (.I(point),
              .O(p));
OR2  XLXI_127 (.I0(LE),
              .I1(XLXN_53),
              .O(g));
OR2  XLXI_128 (.I0(LE),
              .I1(XLXN_52),
              .O(f));
OR2  XLXI_129 (.I0(LE),
              .I1(XLXN_51),
              .O(e));
OR2  XLXI_130 (.I0(LE),
              .I1(XLXN_50),
              .O(d));
OR2  XLXI_131 (.I0(LE),
              .I1(XLXN_49),

```

```

        .O(c));
OR2  XLXI_132 (.I0(LE),
              .I1(XLXN_48),
              .O(b));
OR2  XLXI_133 (.I0(LE),
              .I1(XLXN_47),
              .O(a));
INV  XLXI_134 (.I(D3),
              .O(XLXN_68));
INV  XLXI_135 (.I(D1),
              .O(XLXN_64));
INV  XLXI_138 (.I(D0),
              .O(XLXN_65));
INV  XLXI_139 (.I(D2),
              .O(XLXN_63));
endmodule

```

2. 建立基准测试波形文件

- (a) 在 New Source 中创建文件名为 MyMC14495，勾选 Add to Project
 选择 MyMC14495 模块，点击 Next，在 Summary 窗口再点击 Finish，
 进入 MyMC14495sim.v 编辑窗口

- (b) 仿真激励输入

在源文件中添加以下代码：

```

`timescale 1ns / 1ps
module MyMC14495_MyMC14495_sch_tb();
    reg LE;
    reg point;
    reg D3;
    reg D1;
    reg D0;
    reg D2;
    wire p;
    wire g;
    wire f;
    wire e;
    wire b;
    wire c;
    wire d;
    wire a;
    MyMC14495 UUT (
        .LE(LE),
        .point(point),

```

```
        .p(p),
        .g(g),
        .f(f),
        .e(e),
        .b(b),
        .c(c),
        .d(d),
        .a(a),
        .D3(D3),
        .D1(D1),
        .D0(D0),
        .D2(D2)
    );
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
endmodule
```

(c) 点击 Behavioral Check Syntax，进行激励代码的查验。通过后查看仿真激励波形，结果如下图：

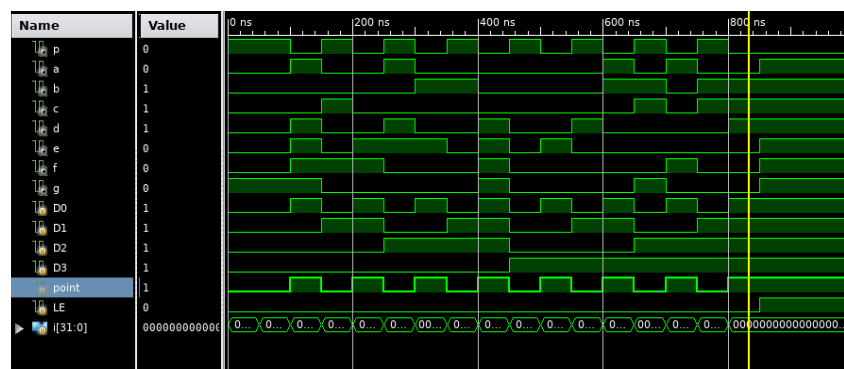


图 3.4 仿真波形图

3. 生成逻辑符号图

(a) Create Schematic Symbol

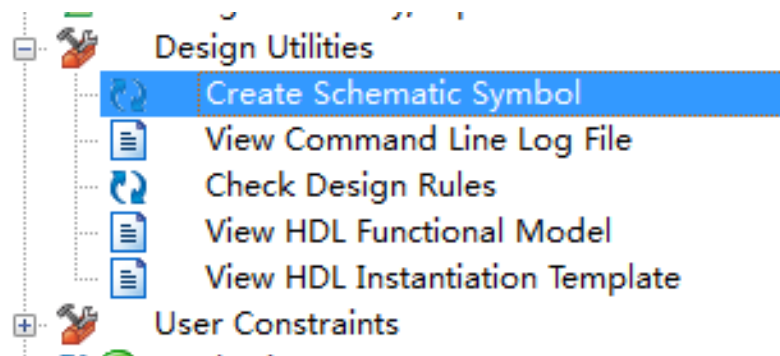


图 3.5 Create Schematic Symbol

(b) 符号图位于工程根目录

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

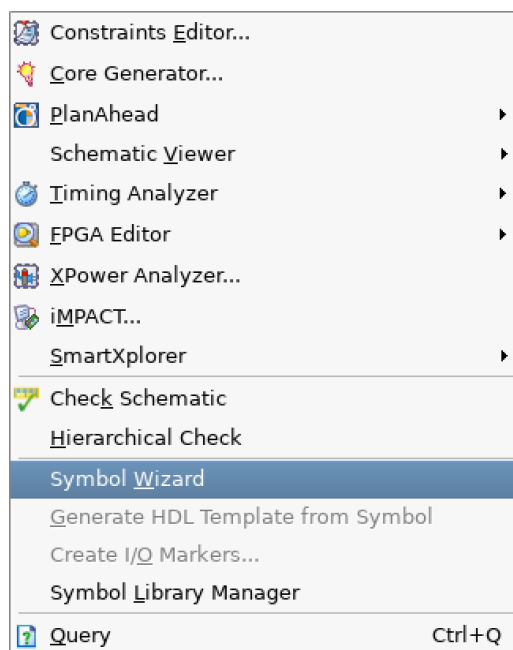


图 3.6 Symbol Wizard

(c) 在新工程中使用，把 .sym 和 .sch 复制到对应工程目录

3.2 实现数码管显示

1. 设计电路

(a) 新建工程 DispNumber_sch

(b) 新建 schematic 文件 DispNumber_sch

(c) 复制 MyMC14495. sym 和 .sch 文件到工程目录并加入工程 (Add Source)

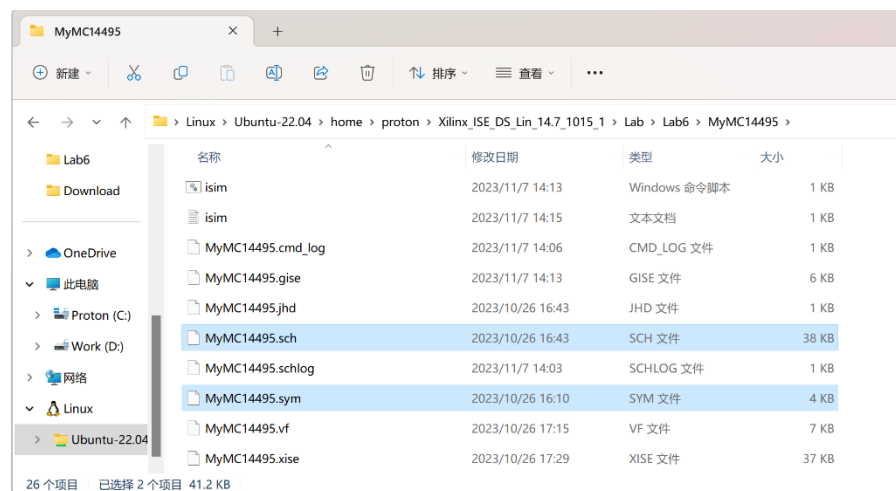


图 3.7 复制 sch 和 sym 文件

(d) 在 Symbols 框里的第一个元件，就是之前设计的 MyMC14495

(e) 根据实验原理，用原理图输入的方式输入如下电路原理图：

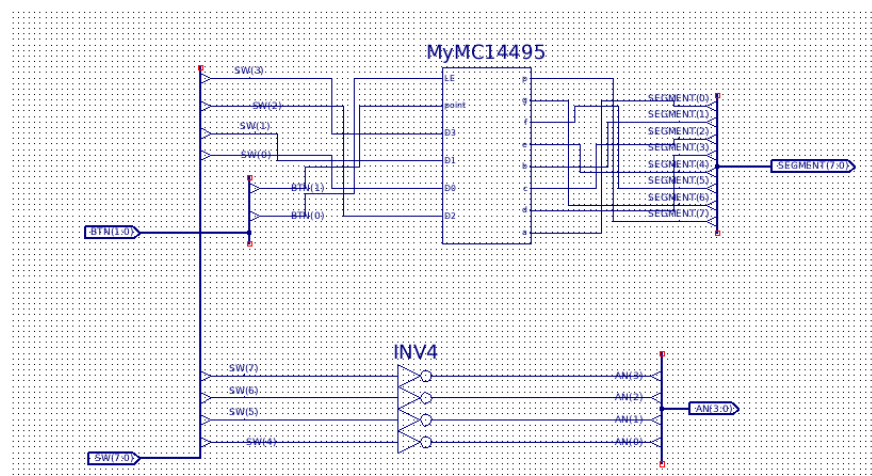


图 3.8 电路原理图

2. 建立 K7 文件

(a) 输入引脚

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

(b) 输出引脚

```
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;

NET "SW[0]" LOC = AA10 | IOSTANDARD = LVCMOS15;#SW[0]
NET "SW[1]" LOC = AB10 | IOSTANDARD = LVCMOS15;#SW[1]
NET "SW[2]" LOC = AA13 | IOSTANDARD = LVCMOS15;#SW[2]
NET "SW[3]" LOC = AA12 | IOSTANDARD = LVCMOS15;#SW[3]
NET "SW[4]" LOC = Y13 | IOSTANDARD = LVCMOS15;#SW[4]
NET "SW[5]" LOC = Y12 | IOSTANDARD = LVCMOS15;#SW[5]
NET "SW[6]" LOC = AD11 | IOSTANDARD = LVCMOS15;#SW[6]
NET "SW[7]" LOC = AD10 | IOSTANDARD = LVCMOS15;#SW[7]

```

3. 下载到 SWORD 板验证

(a) 点击 Generate Programming Files, 通过后点击 Configure Target Device -> Manage Configuration Project, 将文件导出到 SWORD 实验板。

(b) 根据电路原理图和引脚约束文件

- “AF10” 开关对应 “point”, 控制是否显示小数点
- “AF13” 开关对应 “LE”, 控制使能
- “AA10” “AB10” “AA13” “AB12” 四个开关对应 D0, D1, D2, D3, 控制显示数字
- “Y13” “Y12” “AD11” “AD10” 四个开关对应 SW0, SW1, SW2, SW3, 控制对应灯亮灭

(c) 实验结果如下:

- 当 LE = 1 时

此时使能被关闭, 相当于电路被关闭, 由于本电路是共阳电路, 因此每个灯管不会亮起

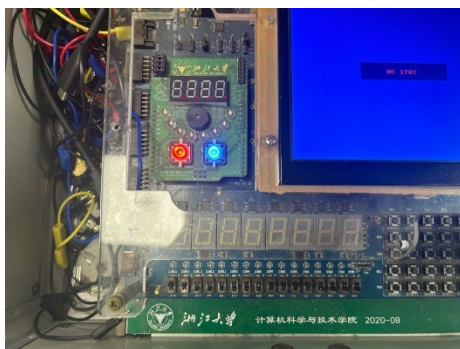


图 3.9 只打开灯控开关



图 3.10 全部打开

由以上两张图可见无论是否打开灯控开关，灯都不会亮灭变化，因此 $LE = 1$ 使得电路断开。但是 point 开关仍会控小数点是否显示，是因为 LE 是能并不会控制 point

➤ 当 $LE = 0$ 时

通过控制从“AA10”到“AD10”8 个开关控制灯的亮灭及显示的数字
测试灯控制开关：

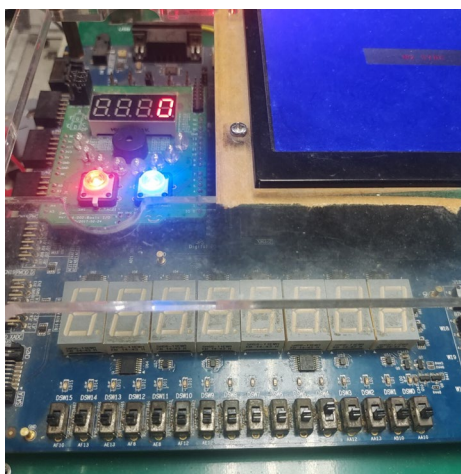


图 3.11 0001

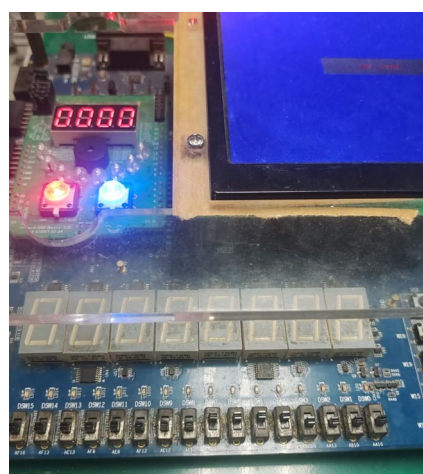


图 3.12 1111

四个灯控制开关全部打开，通过右边四个开关控制显示数字：



图 3.15 0000

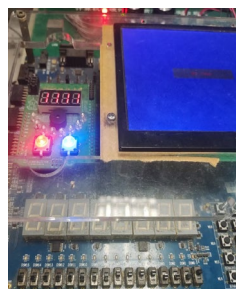


图 3.16 0001

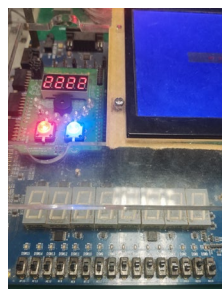


图 3.17 0010

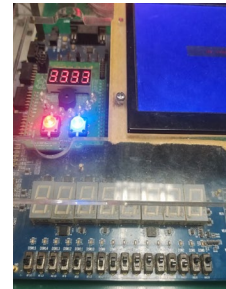


图 3.18 0011

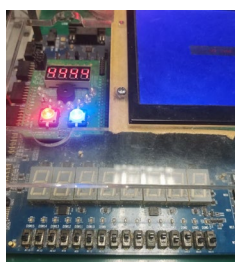


图 3.19 0100

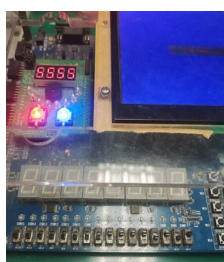


图 3.20 0101

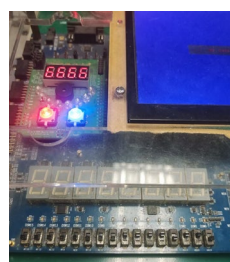


图 3.21 0110

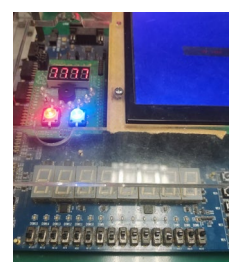


图 3.22 0111

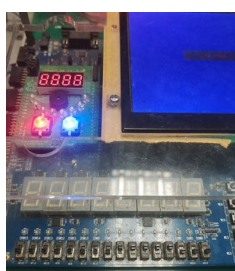


图 3.23 1000

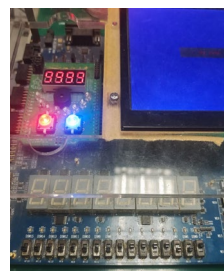


图 3.24 1001

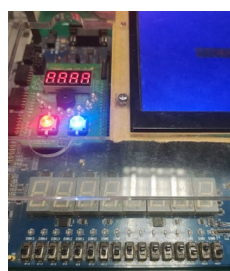


图 3.25 1010

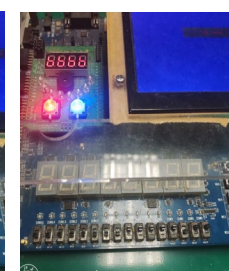


图 3.26 1011

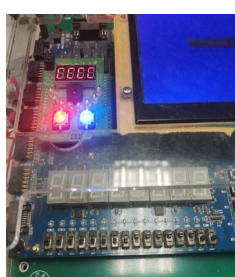


图 3.27 1100

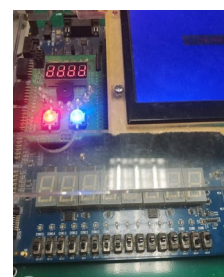


图 3.28 1101

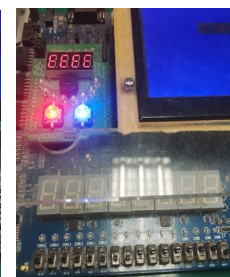


图 3.29 1110

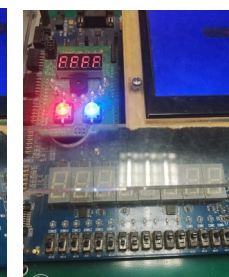


图 3.30 1111

➤ 检查小数点功能

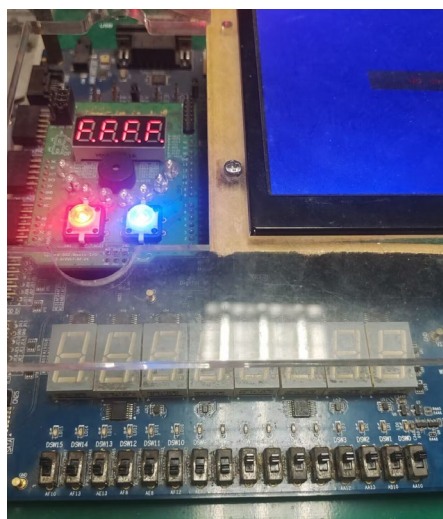


图 3.31 1111

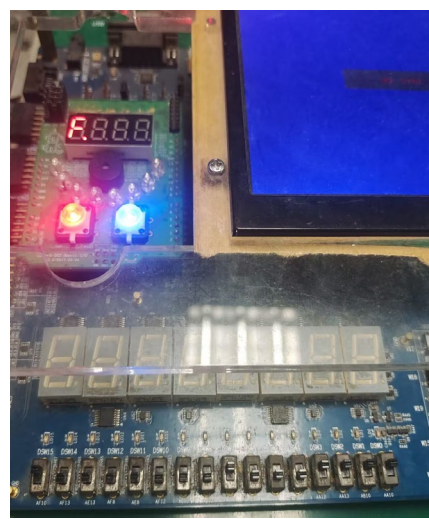


图 3.32 1000

四、实验结果分析

本次实验从七位数码管控制出发，通过变量编码器的应用，依次完成硬件电路图设计、仿真模拟、开发板测试三个步骤

1. 电路图绘制与硬件描述代码

本次实验中所使用的电路图来自课程 PPT，通过电路图生成硬件描述代码后，将代码与图中各组件依次对应，使用了多个逻辑门和反相器(INV)来实现开关信号和 LED 输出之间的逻辑关系：

- (a) INV 用于对输入信号取反
- (b) AND3 用于实现三个输入信号的与逻辑运算
- (c) OR4 用于实现四个输入信号的或逻辑运算。

2. 仿真模拟

- (a) 仿真激励输入的的代码使得 D3,D2,D1,D0 从 0, 0, 0, 0 变到 1, 1, 1, 1，逐次遍历真值表。仿真波形图中对应输出结果与预期相同。
- (b) 在助教老师的指导下，学习判断与分析仿真波形图：

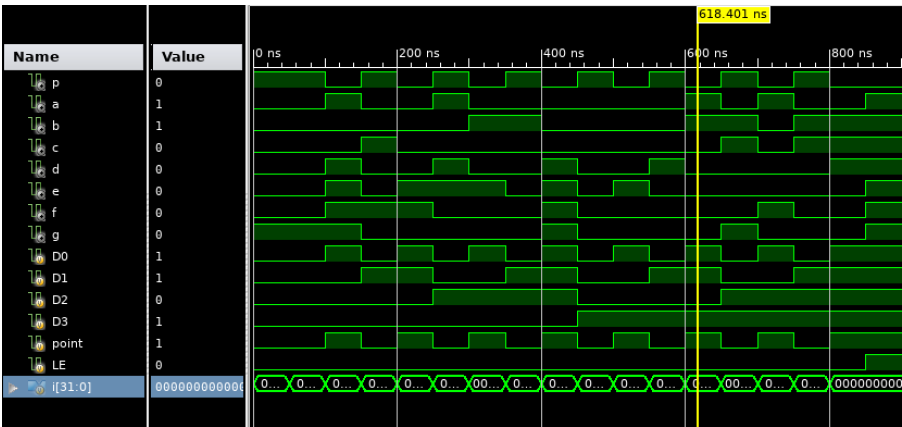


图 4.1 波形图分析

在图 4.1 中，只有 a, b 为 1，其余为 0，对应到字母 b，此时 D0 至 D1 为 1011，对应十六进制的 b，因此仿真波形符合预期

3. 开发板验证

在开发板上，有两个左侧按钮，分别标记为 AF13 和 AF10。这两个按钮的功能是控制数字和小数点的显示开关。另外，还有四个开关，它们分别位于 Y13、Y12、AD1 和 AD10 引脚。这四个开关用于输入二进制码，然

后将其转化为 BCD 码，以便在七段数码管上显示相应的结果。最后，在开发板的右侧，有四个按钮，分别标记为 AA10、AB10、AA13 和 AA12，它们用于输入数据，将数据以 BCD 码的形式显示在七段数码管上。这些按钮和开关的功能在引脚约束文件中都有相应的定义和配置。

五、讨论与心得

Lab6 的实验原理图相对复杂，需要进行一些尺寸的调整以便规整地绘制。如果在绘图过程中出现错误，可以采取以下步骤进行修正：

1. 调整画板尺寸：初始的画板尺寸可能不足以容纳复杂的电路图，因此需要进行适当的尺寸调整，以确保能够绘制清晰的电路图。
2. 检查数字显示问题：如果在物理验证中发现数字显示有误，可以通过对比仿真图来检查相应的电路图模块。确保电路连接正确，没有错误。
3. 使用“branch”查看连接状态：在绘制电路图时，可以选择“branch”来查看连接状态。这会使所有被选中的分支高亮显示（通常是红色），以便更清晰地查看它们之间的连接情况。这有助于确保连接正确无误。
4. 修改时选择“Select Segment”：在进行修改时，可以将选项设置为“Select Segment”，以方便删除或移动线段。这可以使修改变得更加方便和精确。

总结来说，本次实验的难点主要在于绘制复杂的电路图，其他原理和概念都已经学过，在电路图的绘制和仿真过程中，细心和耐心是非常重要的，确保一切都按照预期进行，以获得准确的实验结果。