

浙江大学

本科实验报告

课程名称:	数字逻辑设计
姓 名:	
学 院:	计算机科学与技术学院
专 业:	计算机科学与技术
邮 箱:	
QQ 号:	
电 话:	
指导教师:	蔡铭
报告日期:	2023 年 11 月 18 日

浙江大学实验报告

课程名称：_____ 数字逻辑设计 _____ 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 _____

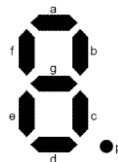
学生姓名：_____ 学号：_____ 同组学生姓名：_____

实验地点：_____ 紫金港东四 509 室 _____ 实验日期：_____ 2023 _____ 年 _____ 11 _____ 月 _____ 1 _____ 日

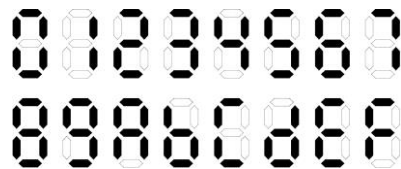
一、操作方法和实验步骤

实验背景

- 七段数码管由 7 个 LED（发光二极管）组成，另外，我们还添加了一个小数点。



- 七段数码管可以用来将输出信号转换为人类能够阅读的“数字”。其中，十六进制下的十六个数字表示为下图的形式。



- 本次实验，我们将设计一个 **译码器**，它将得到的 4 位输入转化为 7 位输出。

设计实现 MC14495 的功能

- 我们将要实现的部件 MyMC14495 的端口如下：

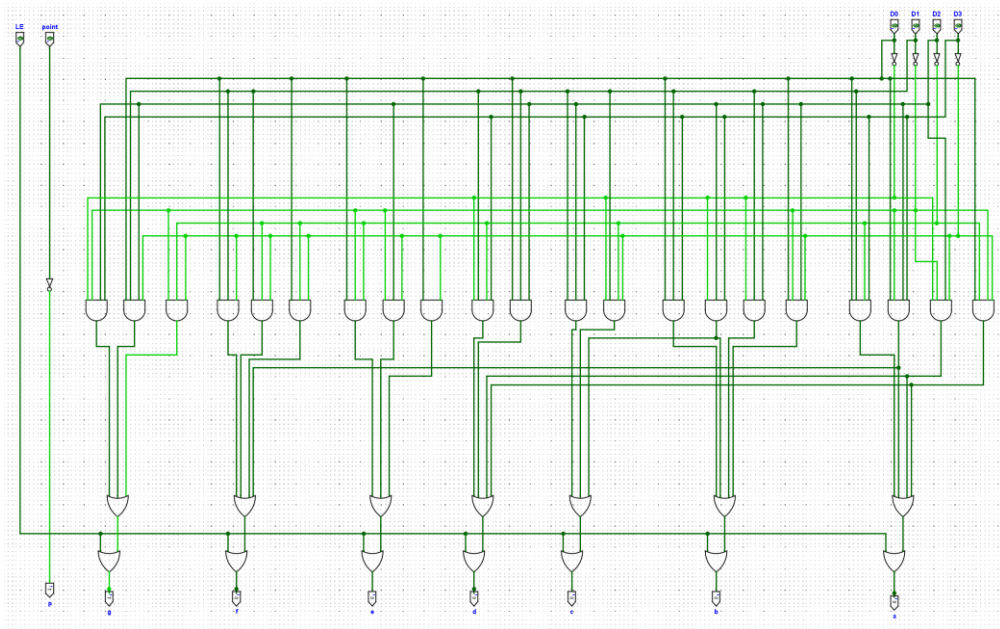
– D3~D0: 输入的 4 位二进制数字

- LE: 使能信号, 低电平有效
- point: 用来表示小数点是否点亮, 高电平有效
- 输出信号 a~g, p: 均为低电平有效

- 根据实验背景中的描述, 我们可以得到如下真值表:

X	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1
1	1	0	0	1	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

- 注意, 这些 LED 共阳极, 即低电平有效, 只有输出为 0 时才有效点亮, 而 point 为高电平有效。
- 我使用了原理图完成了该部件的绘制:



- 接下来, 我们对 MyMC14495 进行仿真
- 以下是仿真激励的代码片段

```
`timescale 1ns / 1ps

module MyMC14495_tb();
```

```

// Inputs
reg [3:0] D;
reg LE;
reg point;

// Output
wire p;
wire a;
wire b;
wire c;
wire d;
wire e;
wire f;
wire g;

// Instantiate the UUT
MyMC14495 MC14495_inst (
    .D0(D[0]),
    .D1(D[1]),
    .D2(D[2]),
    .D3(D[3]),
    .LE(LE),
    .point(point),
    .p(p),
    .a(a),
    .b(b),
    .c(c),
    .d(d),
    .e(e),
    .f(f),
    .g(g)
);

integer i;

initial begin
    D = 4'b0000;
    LE = 0;
    point = 0;
    for (i = 0; i <= 15; i = i + 1)
        begin
            D = i;
            #50;
        end
    #50;
    LE = 1;
end
endmodule

```

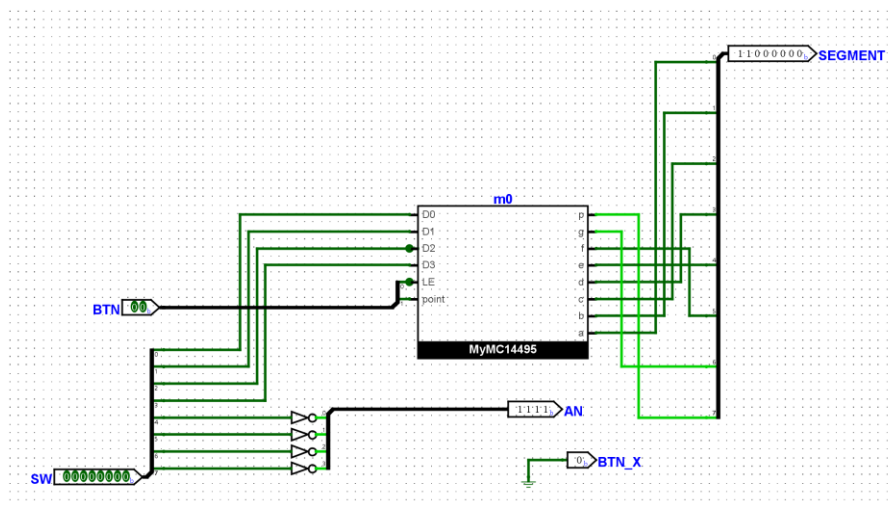
MC14495 的简单使用

- 接下来，我们将上述部件封装并生成 **DispNum** 模块，以下为实际下板过程中的输入和输出：

– 输入端口

- **BTN[1:0]**: 按钮阵列中同一行的两个按钮

- SW[7:0]: 板上右半边的 8 个开关
- 输出端口
 - 七段数码管（含小数点）
 - SEGMENT[7] 与 p 连接
 - SEGMENT[0]~SEGMENT[6] 分别与 a, b, c, d, e, f, g 连接
 - AN[3:0] 用来直接控制 4 个七段数码管的亮灭
 - BTN_X 恒置 0，选择按钮阵列中的某一行
- 以下是用 Logisim 绘制得到的原理图：



- 最后，我们将原理图用 Verilog 导出并导入 Vivado
- 然后，我们需要写入引脚约束文件

```
# Filename: constraints_lab6.xdc
## Constraints file for Lab6

# Switches as inputs
set_property PACKAGE_PIN AA10 [get_ports {SW[0]}]
set_property PACKAGE_PIN AB10 [get_ports {SW[1]}]
set_property PACKAGE_PIN AA13 [get_ports {SW[2]}]
set_property PACKAGE_PIN AA12 [get_ports {SW[3]}]
set_property PACKAGE_PIN Y13 [get_ports {SW[4]}]
set_property PACKAGE_PIN Y12 [get_ports {SW[5]}]
set_property PACKAGE_PIN AD11 [get_ports {SW[6]}]
set_property PACKAGE_PIN AD10 [get_ports {SW[7]}]
set_property IOSTANDARD LVCMOS15 [get_ports {SW[0]}]
set_property IOSTANDARD LVCMOS15 [get_ports {SW[1]}]
set_property IOSTANDARD LVCMOS15 [get_ports {SW[2]}]
set_property IOSTANDARD LVCMOS15 [get_ports {SW[3]}]
set_property IOSTANDARD LVCMOS15 [get_ports {SW[4]}]
set_property IOSTANDARD LVCMOS15 [get_ports {SW[5]}]
set_property IOSTANDARD LVCMOS15 [get_ports {SW[6]}]
set_property IOSTANDARD LVCMOS15 [get_ports {SW[7]}]
```

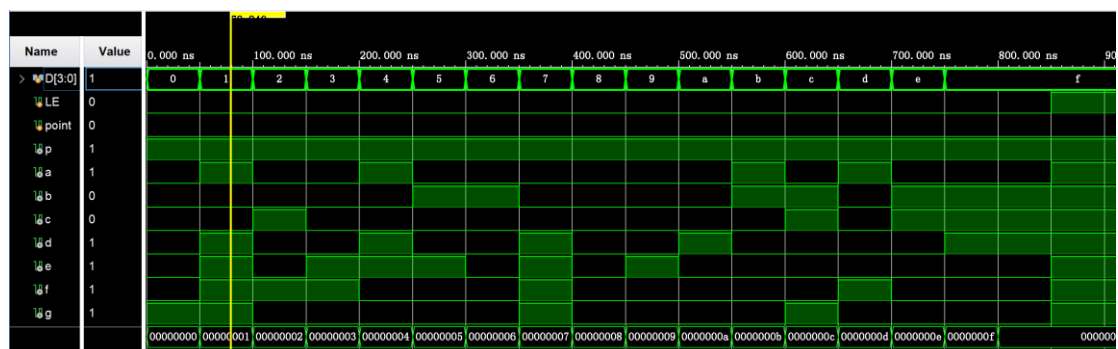
```
# Key as inputs
set_property PACKAGE_PIN V18 [get_ports {BTN[0]}]
set_property IOSTANDARD LVCMOS18 [get_ports {BTN[0]}]
set_property PACKAGE_PIN V19 [get_ports {BTN[1]}]
set_property IOSTANDARD LVCMOS18 [get_ports {BTN[1]}]
set_property PACKAGE_PIN V17 [get_ports {BTN_X}]
set_property IOSTANDARD LVCMOS18 [get_ports {BTN_X}]

# Arduino-Segment & AN
set_property PACKAGE_PIN AD21 [get_ports {AN[0]}]
set_property PACKAGE_PIN AC21 [get_ports {AN[1]}]
set_property PACKAGE_PIN AB21 [get_ports {AN[2]}]
set_property PACKAGE_PIN AC22 [get_ports {AN[3]}]
set_property PACKAGE_PIN AB22 [get_ports {SEGMENT[0]}]
set_property PACKAGE_PIN AD24 [get_ports {SEGMENT[1]}]
set_property PACKAGE_PIN AD23 [get_ports {SEGMENT[2]}]
set_property PACKAGE_PIN Y21 [get_ports {SEGMENT[3]}]
set_property PACKAGE_PIN W20 [get_ports {SEGMENT[4]}]
set_property PACKAGE_PIN AC24 [get_ports {SEGMENT[5]}]
set_property PACKAGE_PIN AC23 [get_ports {SEGMENT[6]}]
set_property PACKAGE_PIN AA22 [get_ports {SEGMENT[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SEGMENT[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SEGMENT[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SEGMENT[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SEGMENT[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SEGMENT[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SEGMENT[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SEGMENT[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SEGMENT[7]}]
```

二、实验结果与分析

对 MC14495 的仿真激励




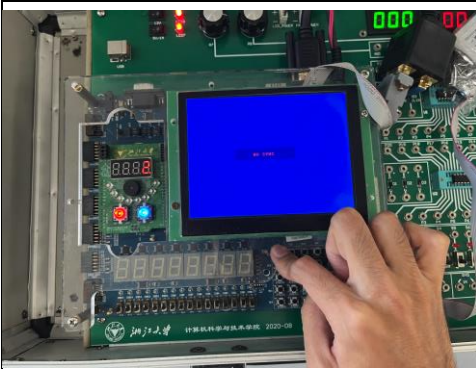

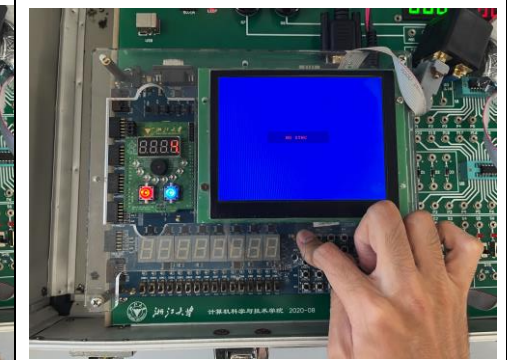
- 以下是仿真激励的结果



- 再次强调，所有控制 LED 的输出都是 低电平有效。

- 当 D 为 0，只有 g 为高电平，其他都是低电平，相当于数码管只有中间的一根不亮，其他都亮，相当于显示 0。
- 当 D 为 1，只有 $b c$ 为低电平，相当于数码管只有最右边的两根亮，相当于显示 1。
- 继续下去，我们将激励结果与真值表进行对比，发现两者完全一致。
- 而对于 LE 使能信号，我们看到它在低电平下才使数码管有效，而它为高电平是，LED 的输出始终为 1，即无效。
- 而 $point$ 为高电平有效。可以看到，在仿真过程中始终为 0，而 p 始终为 1，即不亮。

下板结果

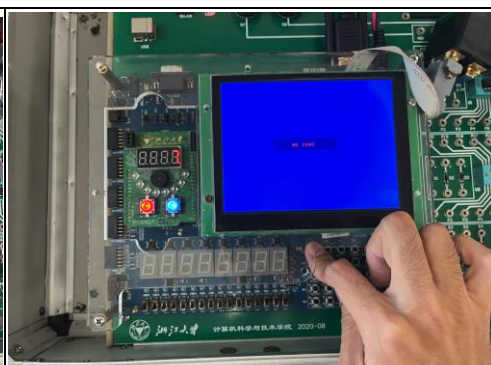
		
<p>控制一个开关开启，亮起一个数码管，按下按钮验证灯管和小数点位的正常显示</p>	<p>开启四个开关，验证四个数码管均可正常亮起</p>	<p>由于之前已经验证了 0，这里我们直接从 1 开始验证。此时，$SW[4:0]=0001$</p>
		
<p>$SW[4:0] = 0010$</p>	<p>$SW[4:0] = 0011$</p>	<p>$SW[4:0] = 0100$</p>



SW[4:0] = 0101



SW[4:0] = 0110



SW[4:0] = 0111



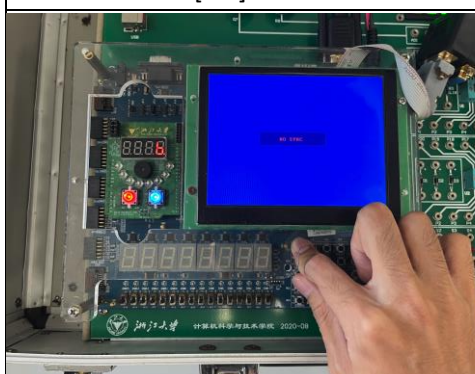
SW[4:0] = 1000



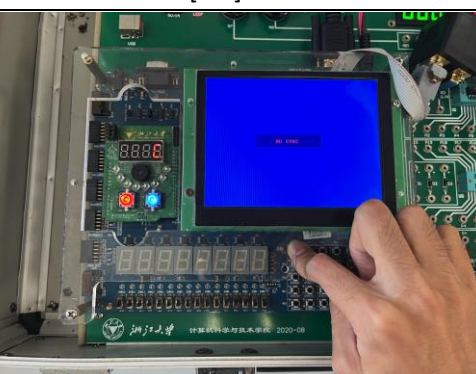
SW[4:0] = 1001



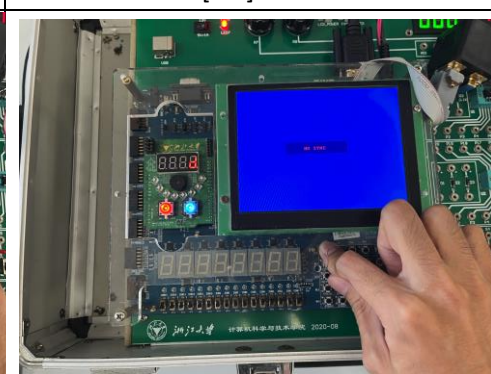
SW[4:0] = 1010



SW[4:0] = 1011



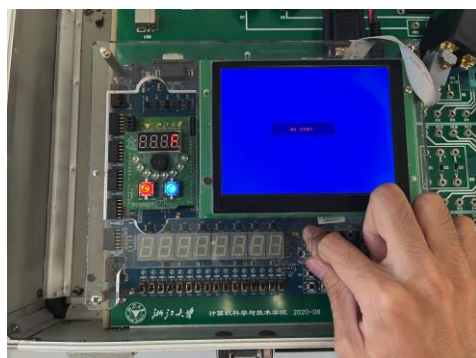
SW[4:0] = 1100



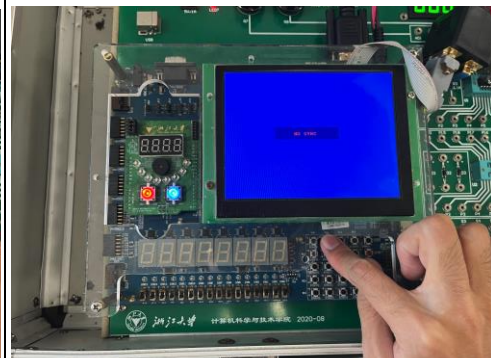
SW[4:0] = 1101



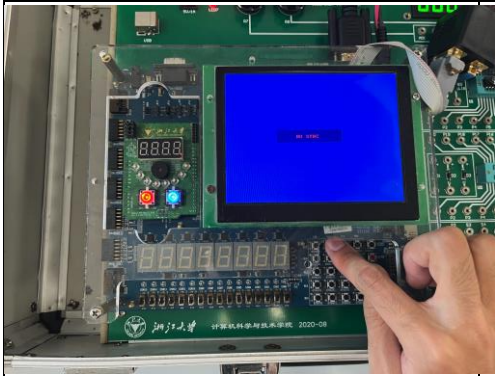
SW[4:0] = 1110



SW[4:0] = 1111



验证 AN 可正常控制亮灭



验证 point 可正常控制小数点位

三、讨论、心得

- 由于之前几次实验我们都是用绘制电路图的方式进行实验的,再加上我还没怎么手写过 **Verilog** 代码,因此这次试验我还是用电路图的方式进行的。而这次电路图应该是几次实验以来最难的,完全没有规律可循,因此在绘制过程中需要更加耐心,并且及时地在 **Logisim** 当中进行调试,验证连线的正确性。
- 在仿真激励的过程中,由于助教给的 **SAMPLE CODE** 是对封装后的部件进行的,而我刚开始还没有封装过,因此在其中遇到了一些波折,后来才研究明白了 **Verilog** 代码中对部件的实例化和引脚表示。
- 在下板的时候,由于我刚开始没有仔细看引脚约束,因此下板的时候根本不知道怎么操作,还以为是电路图画错了,后来才发现是用按钮操纵,花了不少的时间,也终于又看懂了一点按钮的引脚约束应该怎么写。
- 这次试验踩了不少坑,以后的实验过程应该也会相应顺利一些。希望未来的自己越来越好!