编码器译码器设计

# 实验目的

1. 设计4-2优先编码器及译码器，并使它的输出与七段数码管相连，实现优先编码和译码功能。
2. 使用Vivado软件根据提供的DSDB.xdc管脚约束文件，修改上面decoder.v和decoder.xdc程序，将七段数码管改为LED显示，实现二进制编码及译码功能。
3. （附加实验）利用Vivado软件实验实现LCD液晶显示屏Bonus显示功能。

# 实验原理

一、4-2优先编码器及译码器的设计

基于电原理图的组合电路设计流程如下：

1. 对设计目标进行分析，得到输入和输出的逻辑关系并定义输入输出变量。

2. 根据输入变量的个数确定输入信号的组合，并根据输入变量的每个组合推出输出变量的状态，由此推出真值表。

3. 根据真值表推出逻辑表达式，并通过不同的化简方法，和所采用的器件获得相应的最简逻辑表达式。

4. 得出电原理图。

5. 验证电路功能。

二、LCD液晶显示屏显示Bonus

本实验使用基本门电路和Verilog硬件语言设计4-2优先编码器及译码器，流程如上。

对于LCD液晶显示屏Bonus

薄膜晶体管液晶显示器（ Thin Film Transistor-Liquid Crystal Display ，TFT-LCD），可以对屏幕上的各个独立的像素进行控制。图像产生的基本原理很简单：显示屏由许多可以发出任意颜色的像素组成，只要控制各个像素显示相应的颜色就能达到目的。

HV同步模式下，图像的显示只需要行同步信号(hsync)和场同步信号(vsync)来确定显示时序。H就是Horizontal横向，V就是Vertical纵向。

# 三、实验内容

**一、4-2优先编码器及译码器的设计**

根据4-2优先编码器及译码器的输入与输出做真值表如下：

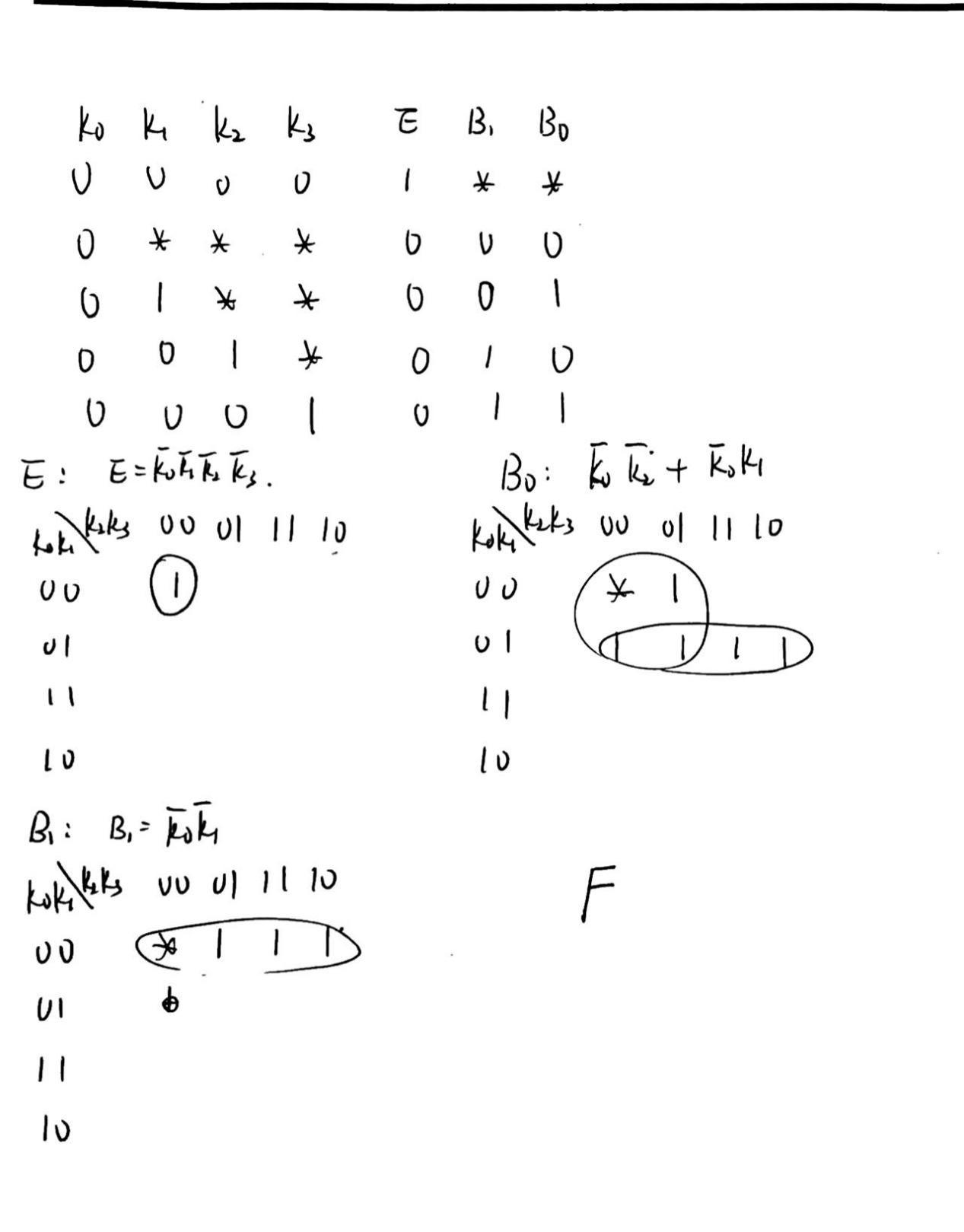


图1 4-2优先编码器及译码器的真值表

其中，作为输入，作为输出，所以有

电路搭建如下：

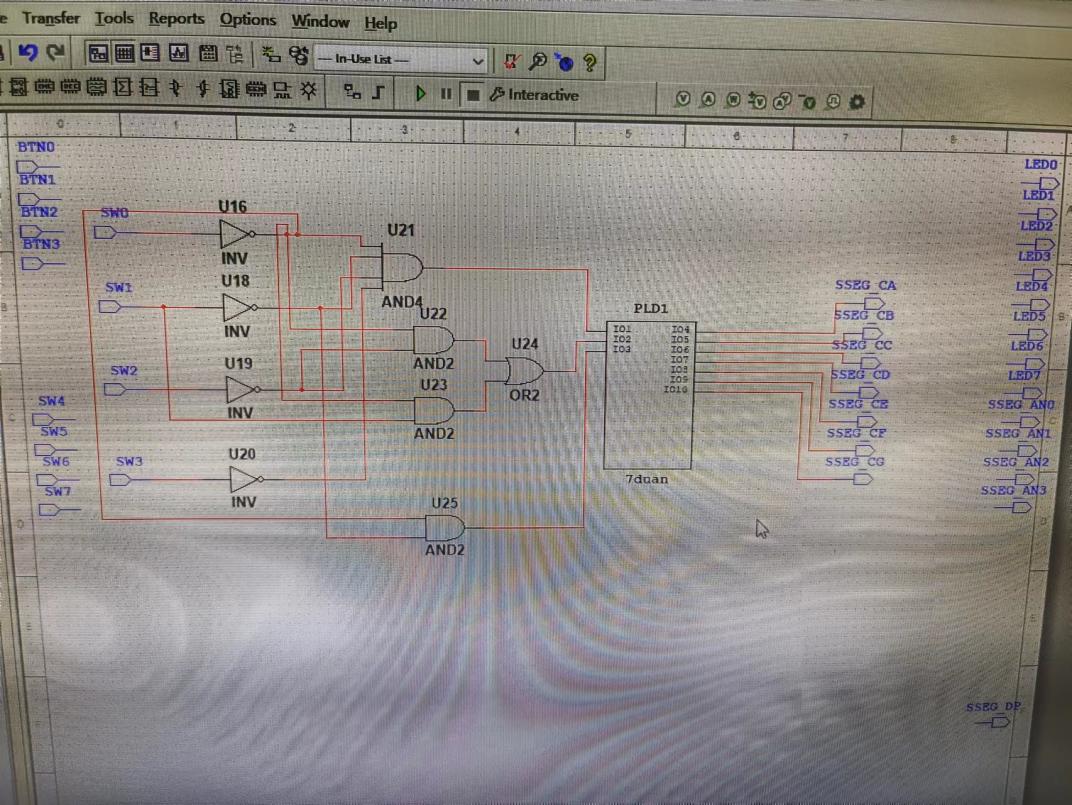


图2 4-2优先编码器及译码器的电路图

其中PLD1为七段数码管电路，如下：

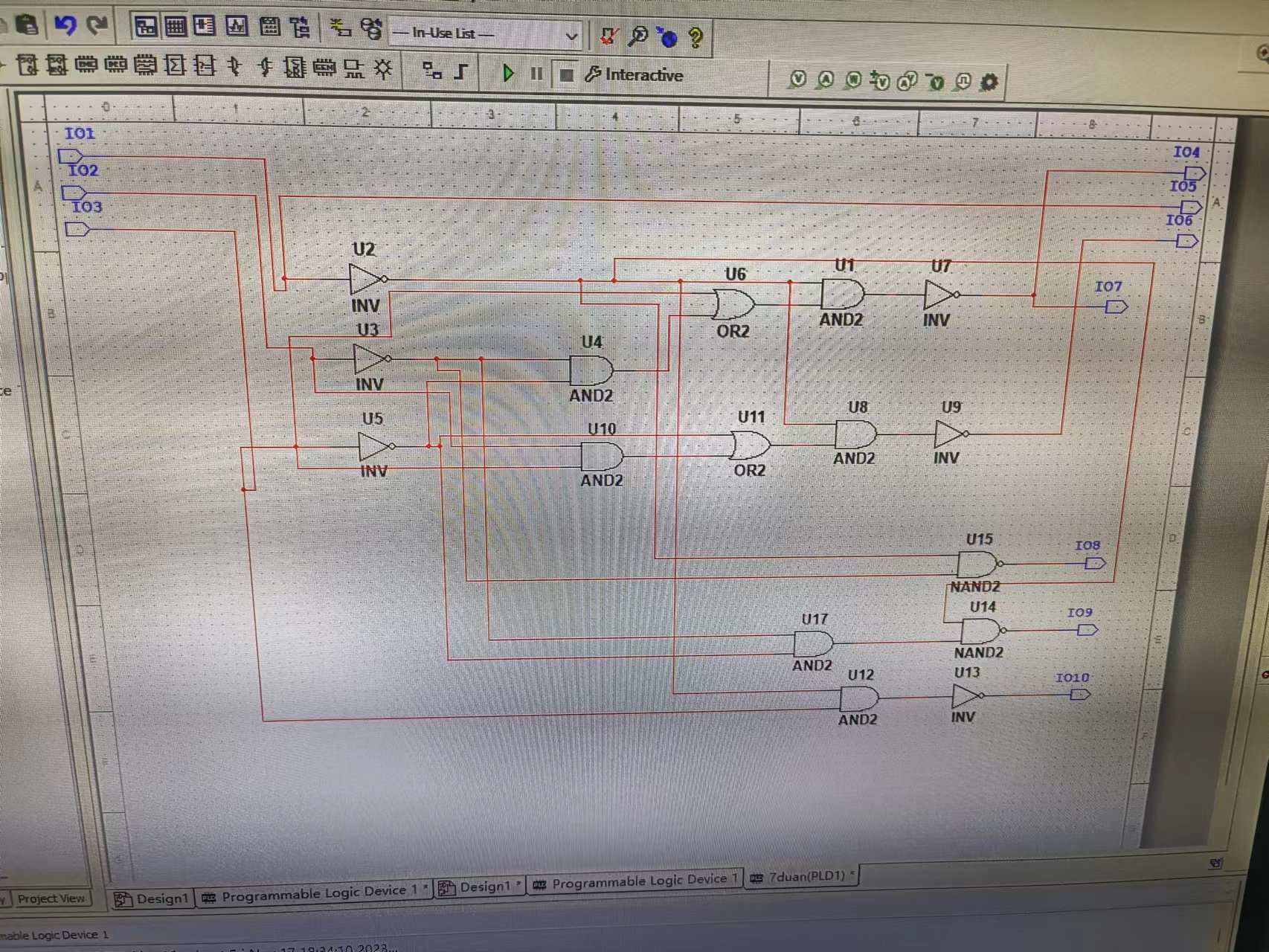


图3 七段数码管的电路图

**二、三位二进制电路Vivado文件的修改**

程序的功能注释：

以下为实现功能的decoder.v文件：

`timescale 1ns / 1ps

module decode (

input wire[2:0] in, **#定义输入变量**

output reg[7:0] led **#定义输出变量**

);

always @(\*) begin **#开始循环**

case(in)  **#判断输入信号**

0: led=8'b00000001; **#输入为0，则第0盏灯亮，以此类推**

1: led=8'b00000010; **#8‘b表示8位二进制，1表示灯亮**

2: led=8'b00000100;

3: led=8'b00001000;

4: led=8'b00010000;

5: led=8'b00100000;

6: led=8'b01000000;

7: led=8'b10000000;

default: led=8'b00000000; **#没有符合以上条件的，则都不亮**

endcase  **#结束判断**

end **#结束循环**

endmodule

以下为实现功能的decoder.xdc文件：

#Switch

set\_property -dict { PACKAGE\_PIN T6 IOSTANDARD LVCMOS33 } [get\_ports { in[0] }];

set\_property -dict { PACKAGE\_PIN U5 IOSTANDARD LVCMOS33 } [get\_ports { in[1] }];

set\_property -dict { PACKAGE\_PIN T4 IOSTANDARD LVCMOS33 } [get\_ports { in[2] }];

#leds

set\_property -dict { PACKAGE\_PIN Y9 IOSTANDARD LVCMOS33 } [get\_ports { led[0] }];

set\_property -dict { PACKAGE\_PIN Y8 IOSTANDARD LVCMOS33 } [get\_ports { led[1] }];

set\_property -dict { PACKAGE\_PIN V7 IOSTANDARD LVCMOS33 } [get\_ports { led[2] }];

set\_property -dict { PACKAGE\_PIN W7 IOSTANDARD LVCMOS33 } [get\_ports { led[3] }];

set\_property -dict { PACKAGE\_PIN V10 IOSTANDARD LVCMOS33 } [get\_ports { led[4] }];

set\_property -dict { PACKAGE\_PIN W12 IOSTANDARD LVCMOS33 } [get\_ports { led[5] }];

set\_property -dict { PACKAGE\_PIN W11 IOSTANDARD LVCMOS33 } [get\_ports { led[6] }];

set\_property -dict { PACKAGE\_PIN V8 IOSTANDARD LVCMOS33 } [get\_ports { led[7] }];

根据DSDB.xdc文件确定输入和输出管脚。

**三、LCD液晶显示屏Bonus（附加实验）**

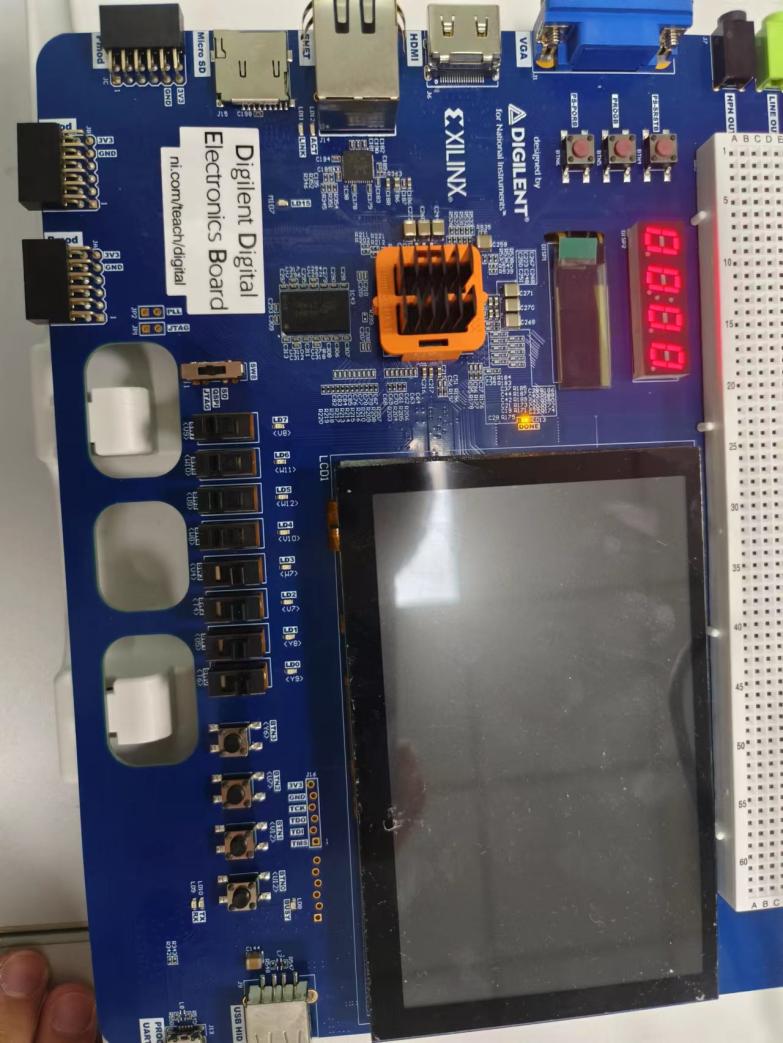
导入decoder.v和decoder.xdc程序，并导入clock wizard库函数，修改时钟频率，编译并运行程序。

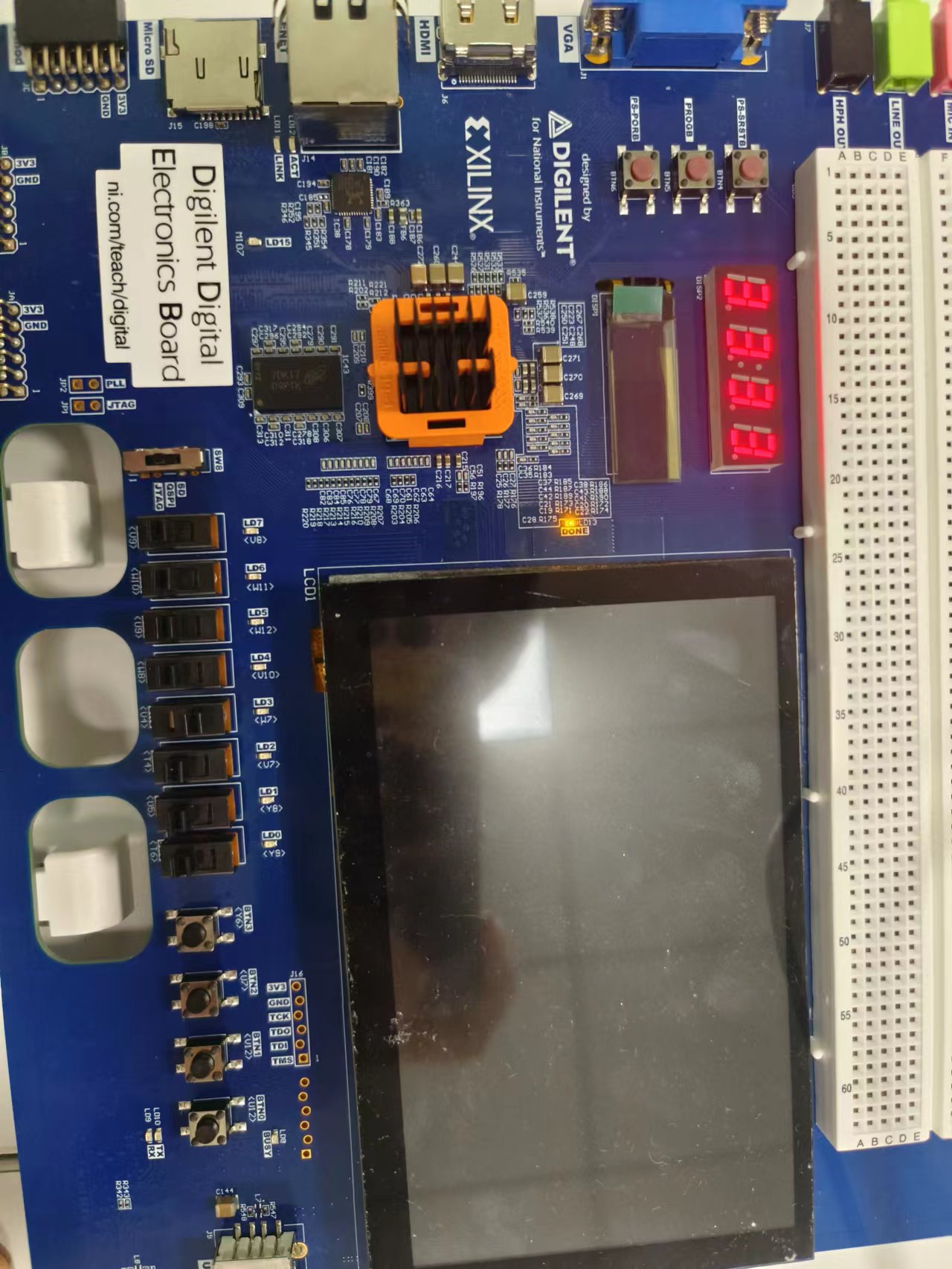
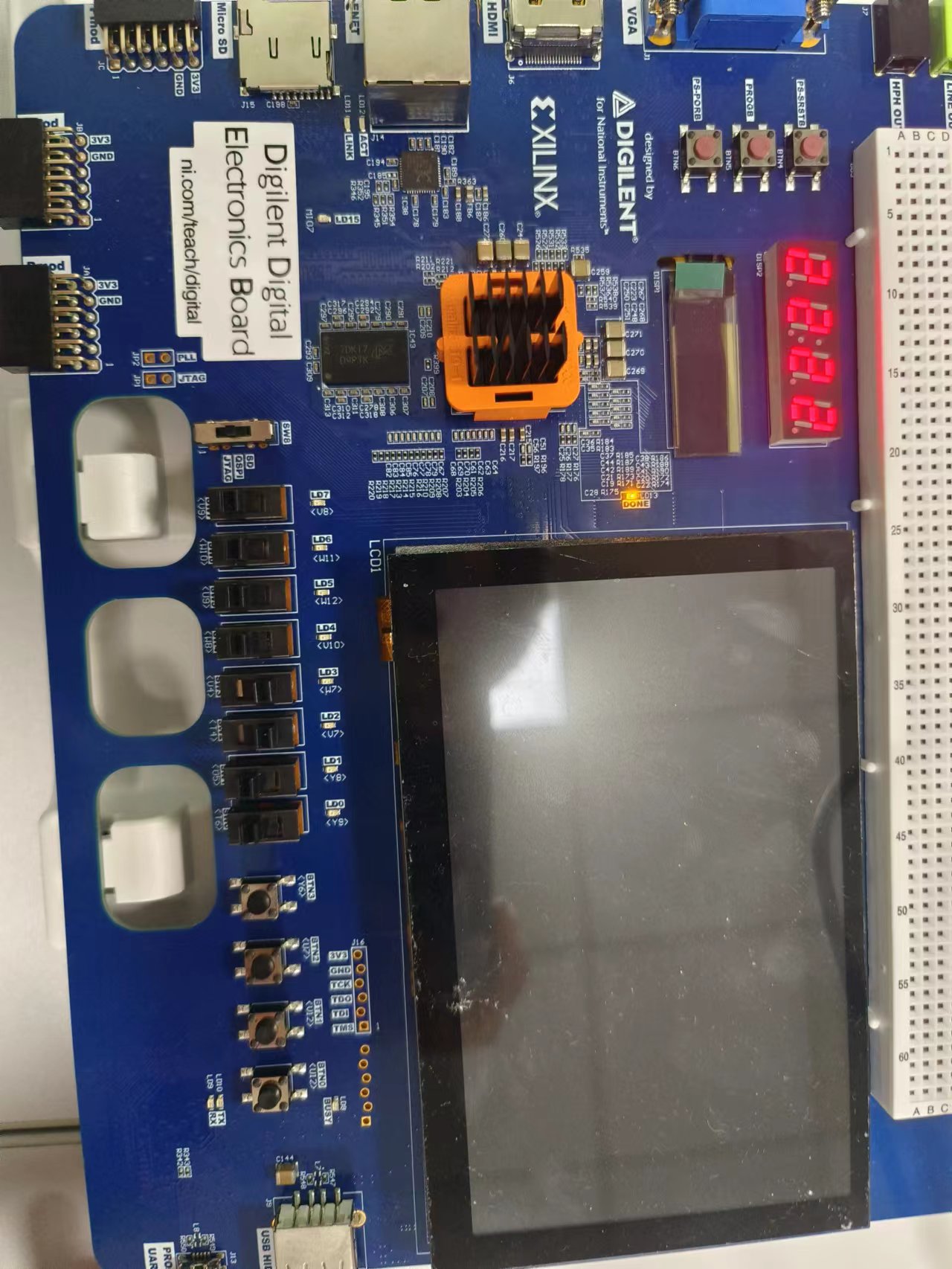
# 四、实验结果及分析

**一、4-2优先编码器及译码器的设计**

实验结果如下：

****

****

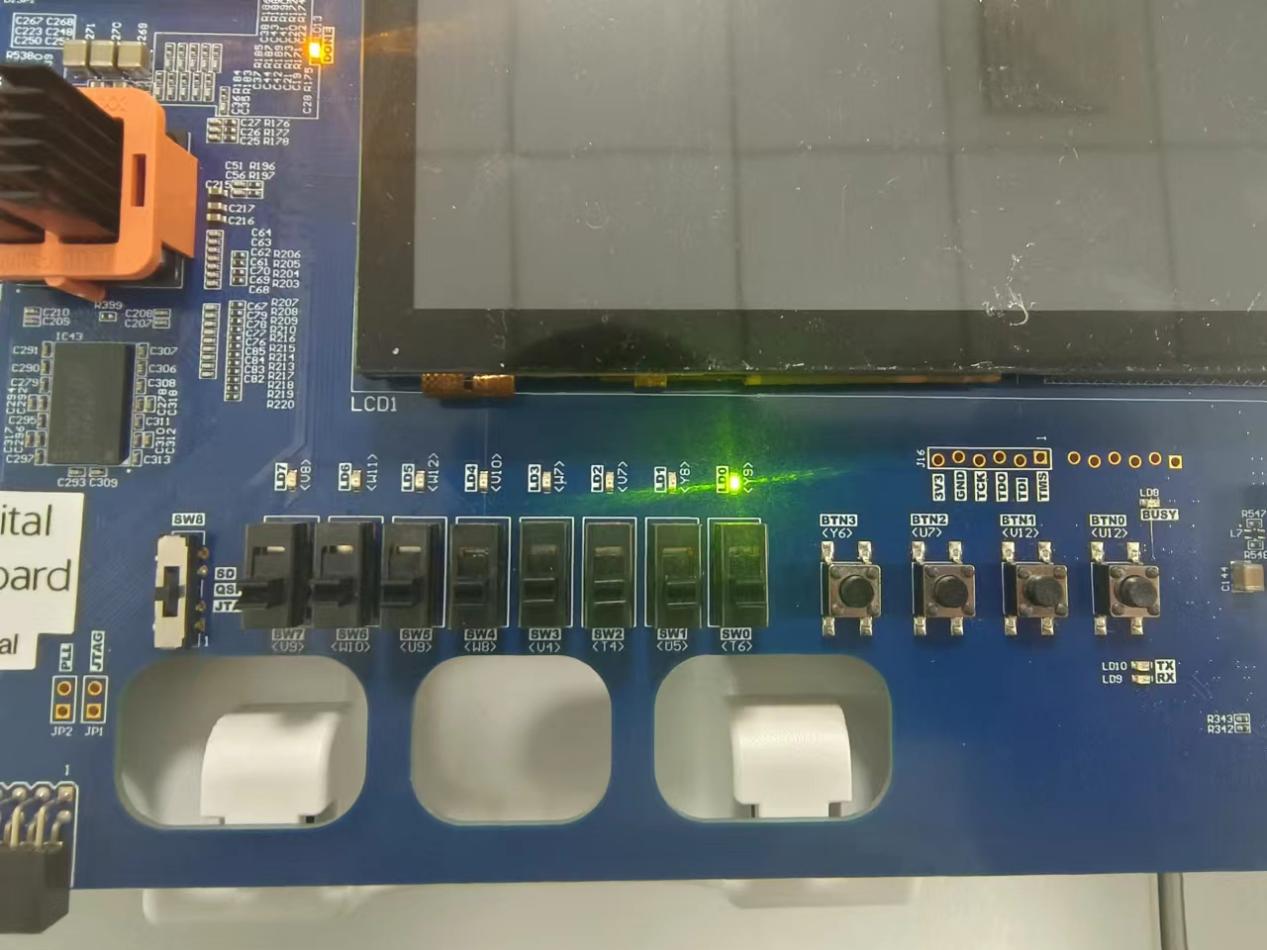
****

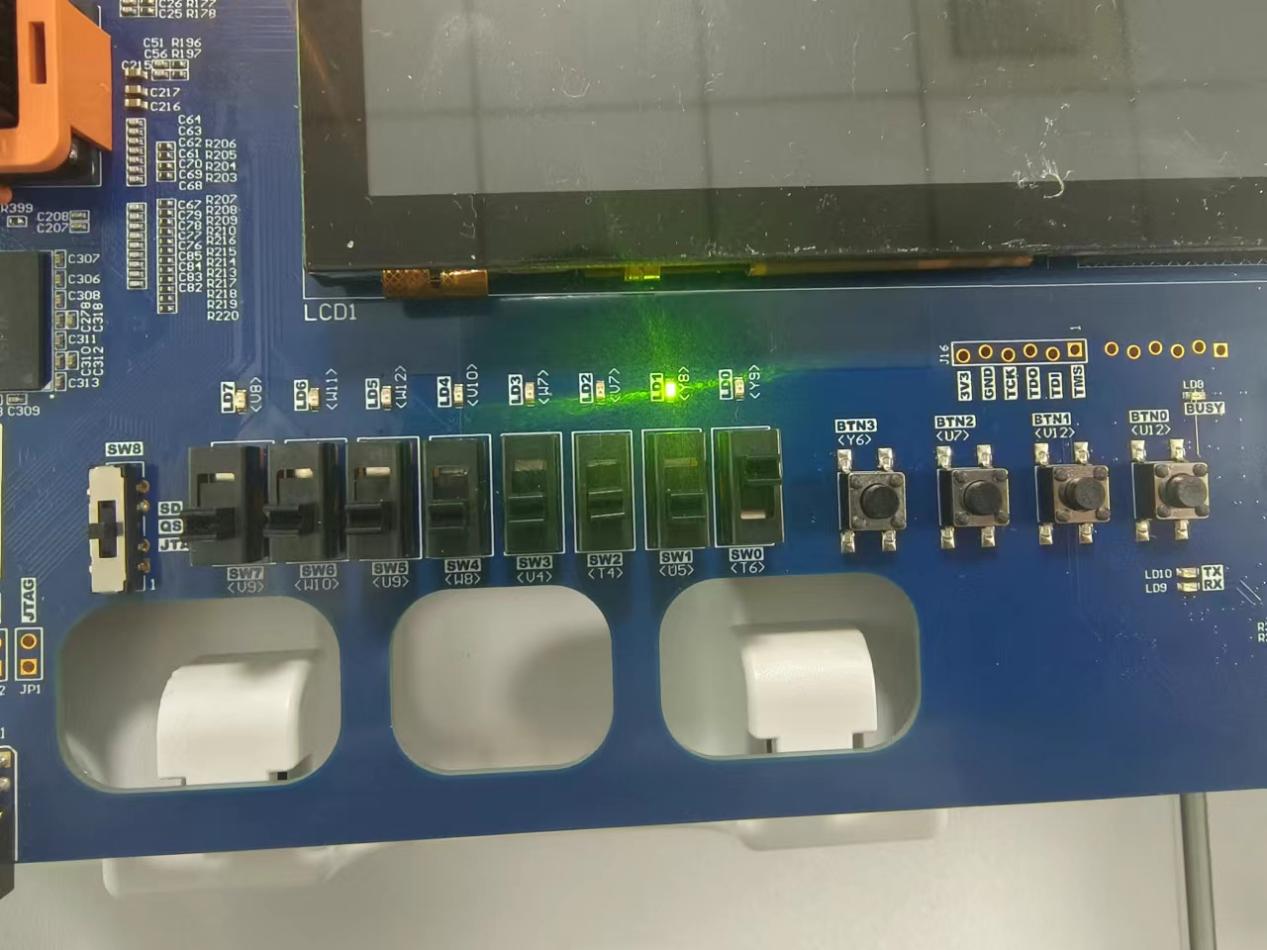
实验结果分析：

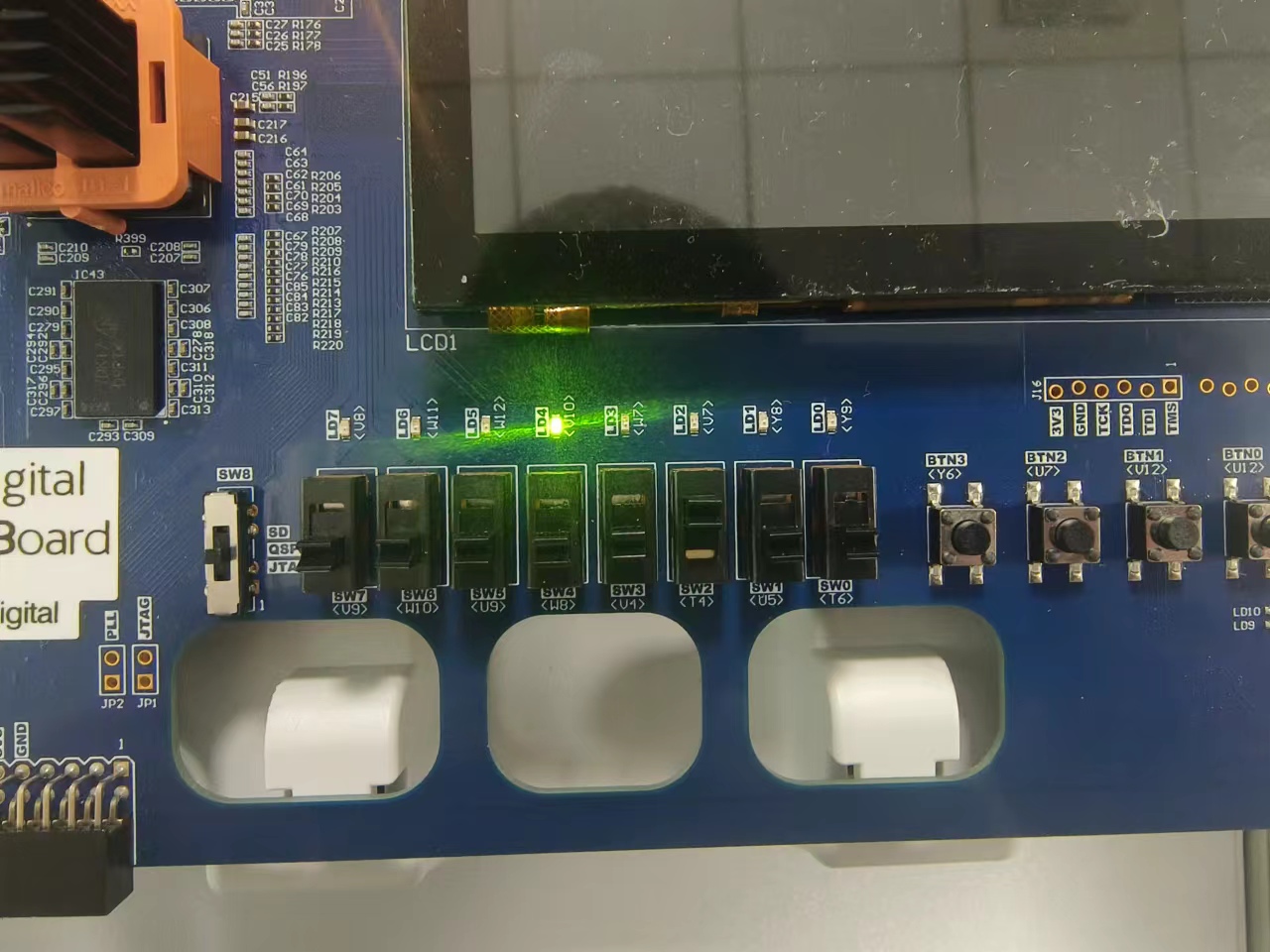
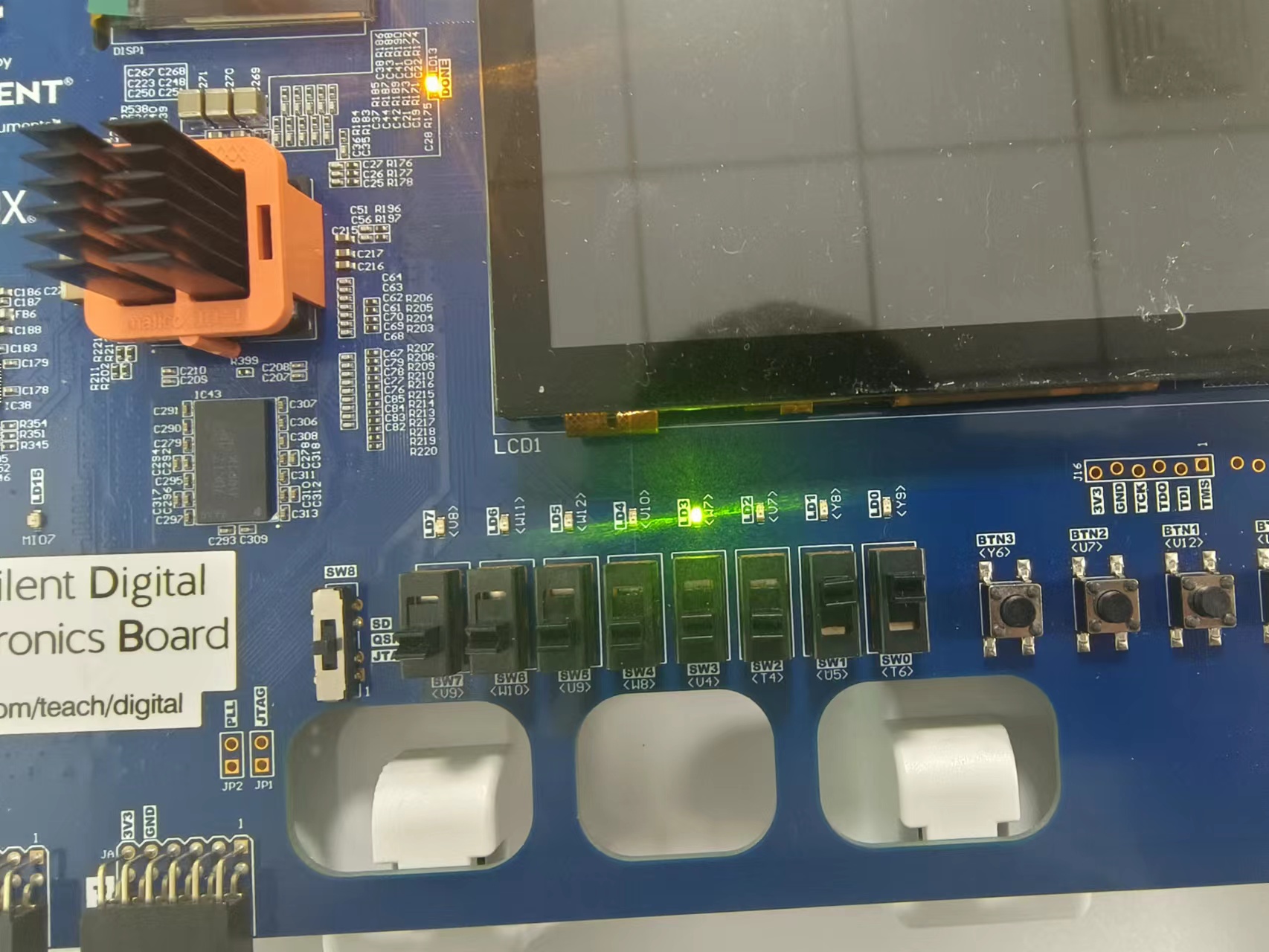
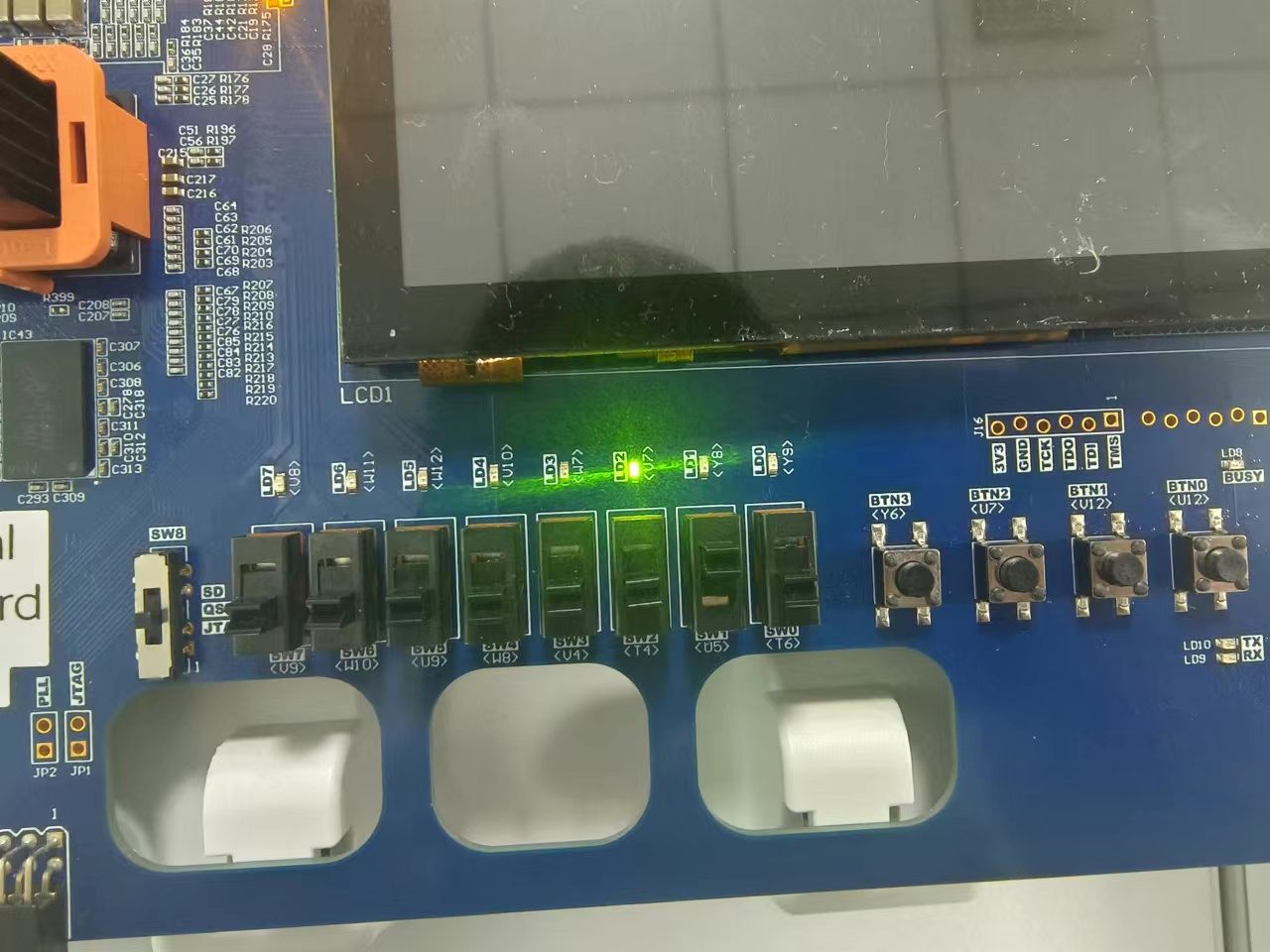
实验中当SW0=0，SW1=0，SW2=0，SW3=0时，四个七段数码管都不亮；当SW0=1，SW1=任意值，SW2=任意值，SW3=任意值时，第0个七段数码管显示0，其余三个七段数码管不亮；当SW0=0，SW1=1，SW2=任意值，SW3=任意值时，第0个七段数码管显示1，其余三个七段数码管不亮；当SW0=0，SW1=0，SW2=1，SW3=任意值时，第0个七段数码管显示2，其余三个七段数码管不亮；当SW0=0，SW1=0，SW2=0，SW3=1时，第0个七段数码管显示3，其余三个七段数码管不亮。实验成功。

**二、三位二进制电路Vivado文件的修改**

实验结果如下：

****

****

****

## 13351ec3d481fbb43a32277c6cfc233f9ac2c02cbd56337c6f2992e9f6c5d9704710acab569716afd00244b48d9bb

实验结果分析：

当SW2，SW1，SW0=000，LED0亮；

当SW2，SW1，SW0=001，LED1亮；

当SW2，SW1，SW0=010，LED2亮；

当SW2，SW1，SW0=011，LED3亮；

当SW2，SW1，SW0=100，LED4亮；

当SW2，SW1，SW0=101，LED5亮；

当SW2，SW1，SW0=110，LED6亮；

当SW2，SW1，SW0=111，LED7亮；

综上，实验成功。

## 三、LCD液晶显示屏Bonus（附加实验）

实验结果如下：



实验结果分析：

附加实验成功。

# 五、遇到的问题

1.在附加实验中遇到了程序无法编译的问题，询问老师后发现程序文件的路径不能有中文。

解决：重新下载文件，跟着视频再做一次。

1. 修改的程序编译发生报错，后发现为defaul未与case各情况对齐。解决：将default与情况1-7的语句对齐。

# 六、反思与心得

这次数电实验让我收获颇丰。我不仅深入理解了优先编码器编码器、译码器的工作原理，还通过实际操作提高了自己的动手能力，锻炼了理论知识的应用能力。同时学习了Vivado软件的编程与运用。我相信这些经验将有助于我未来的学习和工作。