数字电子技术

实验报告

【实验 02-七段 LED 数码管】

上海财经大学 信息管理与工程学院

学生姓名：吉宁岳

学号：2022111899

日期：2022 年 9 月 17 日

一、实验内容

1．设计驱动 4 个七段数码管，显示固定数据：2022。

2．扩展编程，设计驱动 4 个七段数码管，显示内容可以根据 16 个拨码开关(SW) 进行相应的 16 进制数据显示

二、实验方案

整体思路：分时复用七段数码管

参考示例工程：7SegmentLED，Dec7Seg，x7Seg

PART I

在顶层文件中预写入"'h2022"并将其分配给 HEX 值 x，随后将其实例化输出

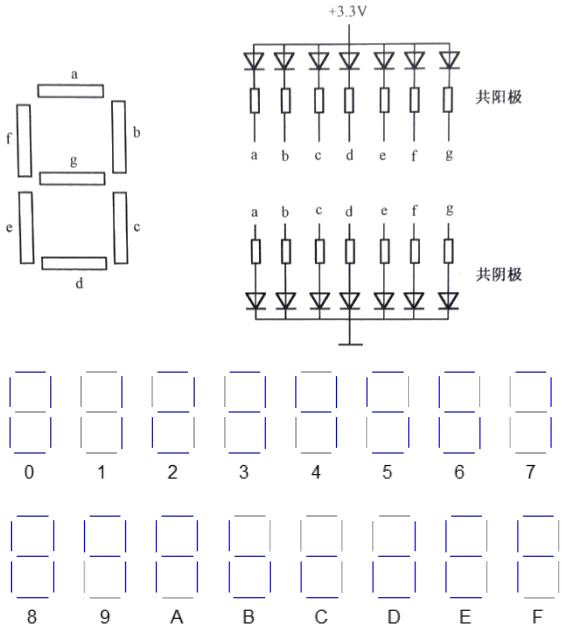
PART II

将拨码开关直接分配给 HEX 值 x，随后将其实例化输出

三、实验分析

1. Dec7Seg.sv（HEX 值→七段数码管实际输出）

```
3  module Dec7Seg(
4      input   logic [3:0] x,
5      output  logic [6:0] a2g );
6
7      assign AN = 4'b0000;
8      assign DP = 1;
9
10     always_comb
11         case (x)
12             'h0: a2g = 7'b0000001;
13             'h1: a2g = 7'b1001111;
14             'h2: a2g = 7'b0010010;
15             'h3: a2g = 7'b0000110;
16             'h4: a2g = 7'b1001100;
17             'h5: a2g = 7'b0100100;
18             'h6: a2g = 7'b0100000;
19             'h7: a2g = 7'b0001111;
20             'h8: a2g = 7'b0000000;
21             'h9: a2g = 7'b0000100;
22             'ha: a2g = 7'b0001000;
23             'hb: a2g = 7'b1100000;
24             'hc: a2g = 7'b0110001;
25             'hd: a2g = 7'b1000010;
26             'he: a2g = 7'b0110000;
27             'hf: a2g = 7'b0111000;
28         default: a2g = 7'b0000001; //0
29         endcase
30 endmodule
```



分析：

1.每一位七段数码管输出内容对应 4 位 BIN 值（即能表达 0-15 的数，和一位 HEX 值相当），故定义 input logic [3:0] x

2.实际输出的 a2g 在约束文件中定义对应数码管，此处 output logic [6:0] a2g

3.a2g 部分由一串七位 BIN 值控制，每一位分别对应图中的 abcdefg 段数码管

4.basys3 板载数码管使用共阳极引脚，因此低电平 0 点亮，高电平 1 熄灭

2. assistDesign.sv（建立分时复用机制）

【这里基本直接引用了 x7Seg 示例工程中的 x7Seg.sv 代码部分】

```
module assistDesign(
                input  logic [15:0] x,
                input  logic        clk,
                input  logic        clr,
                output logic [6:0]  a2g,
                output logic [3:0]  AN, //数码管使能
                output logic        dp ); //小数点

    logic [1:0] s;     //选择哪个数码管
    logic [3:0] digit;
    logic [19:0] clkdiv;

    assign dp = 1;           // DP off
    assign s = clkdiv[19:18];// count every 10.4ms

    //4个数码管 4选1 (MUX44)
    always_comb
        case(s)
            0:  digit = x[3:0];
            1:  digit = x[7:4];
            2:  digit = x[11:8];
            3:  digit = x[15:12];
            default: digit = x[3:0];
        endcase

    //4个数码管的使能
    always_comb
        case(s)
            0:  AN = 4'b1110;
            1:  AN = 4'b1101;
            2:  AN = 4'b1011;
            3:  AN = 4'b0111;
            default: AN = 4'b1110;
        endcase

    // 时钟分频器（20位二进制计数器）
    always @(posedge clk, posedge clr)
      if(clr == 1) clkdiv <= 0;
      else         clkdiv <= clkdiv + 1;

    //实例化 7段数码管
    Dec7Seg s7(.x(digit), .a2g(a2g));
endmodule
```

分析：

1.每个数字分别显示，但是刷新频率极高（190Hz），从而在视觉上实际同时显示

2.本质上是是利用时钟分频器的自动分时复用代替了拨码开关控制的手动复用

3.由于是共阳极数码管，所以整体的点亮熄灭由 AN 引脚控制，AN 由一个四位 BIN 值控制，输入高电平 1 熄灭，低电平 0 点亮

4.通过末句实例化调用 Dec7Seg 来实现 HEX 值→七段数码管实际输出的转换

3. mainDesign.sv（顶层文件）

【参考 x7Seg 示例工程中的 x7Seg_Top.sv 部分】

```
module mainDesign(
    input  logic CLK100MHZ,
    output logic [6:0] a2g,
    output logic [3:0] AN,
    output logic DP );

    logic [15:0] x;
    assign x = 'h2022;

    assistDesign X7(.x(x),
                    .clk(CLK100MHZ),
                    .a2g(a2g),
                    .AN(AN),
                    .dp(DP));
endmodule
```

左图：固定显示 2022；

```
module mainDesign(
    input  logic CLK100MHZ,
    input  logic [15:0] SW,
    output logic [6:0] a2g,
    output logic [3:0] AN,
    output logic [15:0] LED,
    output logic DP );

    logic [15:0] x;
    assign LED = SW;
    assign x = SW;

    assistDesign X7(.x(x),
                    .clk(CLK100MHZ),
                    .a2g(a2g),
                    .AN(AN),
                    .dp(DP));
endmodule
```

右图：根据拨码开关显示

4. Basys3_Master.xdc（约束文件）

```
## Clock signal
set_property PACKAGE_PIN W5 [get_ports CLK100MHZ]
    set_property IOSTANDARD LVCMOS33 [get_ports CLK100MHZ]
    create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports CLK100MHZ]
## Switches
set_property PACKAGE_PIN V17 [get_ports {SW[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {SW[0]}]
set_property PACKAGE_PIN V16 [get_ports {SW[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {SW[1]}]
set_property PACKAGE_PIN W16 [get_ports {SW[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {SW[2]}]
set_property PACKAGE_PIN W17 [get_ports {SW[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {SW[3]}]
set_property PACKAGE_PIN W15 [get_ports {SW[4]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {SW[4]}]
set_property PACKAGE_PIN V15 [get_ports {SW[5]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {SW[5]}]
set_property PACKAGE_PIN W14 [get_ports {SW[6]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {SW[6]}]
set_property PACKAGE_PIN W13 [get_ports {SW[7]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {SW[7]}]
set_property PACKAGE_PIN V2 [get_ports {SW[8]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {SW[8]}]
set_property PACKAGE_PIN T3 [get_ports {SW[9]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {SW[9]}]
set_property PACKAGE_PIN T2 [get_ports {SW[10]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {SW[10]}]
set_property PACKAGE_PIN R3 [get_ports {SW[11]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {SW[11]}]
set_property PACKAGE_PIN W2 [get_ports {SW[12]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {SW[12]}]
set_property PACKAGE_PIN U1 [get_ports {SW[13]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {SW[13]}]
set_property PACKAGE_PIN T1 [get_ports {SW[14]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {SW[14]}]
set_property PACKAGE_PIN R2 [get_ports {SW[15]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {SW[15]}]
## LEDs
set_property PACKAGE_PIN U16 [get_ports {LED[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED[0]}]
set_property PACKAGE_PIN E19 [get_ports {LED[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED[1]}]
set_property PACKAGE_PIN U19 [get_ports {LED[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED[2]}]
set_property PACKAGE_PIN V19 [get_ports {LED[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED[3]}]
set_property PACKAGE_PIN W18 [get_ports {LED[4]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED[4]}]
set_property PACKAGE_PIN U15 [get_ports {LED[5]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED[5]}]
set_property PACKAGE_PIN U14 [get_ports {LED[6]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED[6]}]
set_property PACKAGE_PIN V14 [get_ports {LED[7]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED[7]}]
set_property PACKAGE_PIN V13 [get_ports {LED[8]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED[8]}]
set_property PACKAGE_PIN V3 [get_ports {LED[9]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED[9]}]
set_property PACKAGE_PIN W3 [get_ports {LED[10]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED[10]}]
set_property PACKAGE_PIN U3 [get_ports {LED[11]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED[11]}]
set_property PACKAGE_PIN P3 [get_ports {LED[12]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED[12]}]
set_property PACKAGE_PIN N3 [get_ports {LED[13]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED[13]}]
set_property PACKAGE_PIN P1 [get_ports {LED[14]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED[14]}]
set_property PACKAGE_PIN L1 [get_ports {LED[15]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {LED[15]}]
##7 segment display
set_property PACKAGE_PIN W7 [get_ports {a2g[6]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {a2g[6]}]
set_property PACKAGE_PIN W6 [get_ports {a2g[5]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {a2g[5]}]
set_property PACKAGE_PIN U8 [get_ports {a2g[4]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {a2g[4]}]
set_property PACKAGE_PIN V8 [get_ports {a2g[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {a2g[3]}]
set_property PACKAGE_PIN U5 [get_ports {a2g[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {a2g[2]}]
set_property PACKAGE_PIN V5 [get_ports {a2g[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {a2g[1]}]
set_property PACKAGE_PIN U7 [get_ports {a2g[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {a2g[0]}]
set_property PACKAGE_PIN V7 [get_ports DP]
    set_property IOSTANDARD LVCMOS33 [get_ports DP]
set_property PACKAGE_PIN U2 [get_ports {AN[0]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {AN[0]}]
set_property PACKAGE_PIN U4 [get_ports {AN[1]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {AN[1]}]
set_property PACKAGE_PIN V4 [get_ports {AN[2]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {AN[2]}]
set_property PACKAGE_PIN W4 [get_ports {AN[3]}]
    set_property IOSTANDARD LVCMOS33 [get_ports {AN[3]}]
```

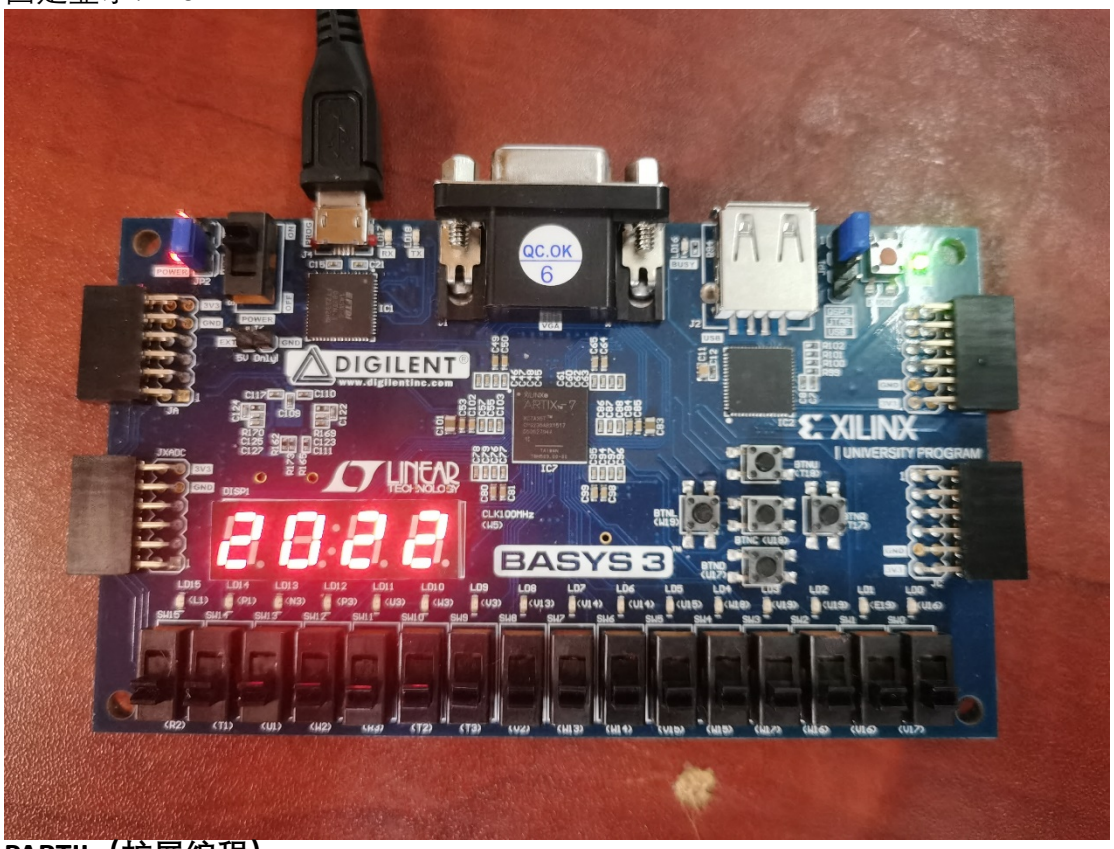
使用模块：时钟信号，拨码开关

LED 灯七段数码管

四、实验结果

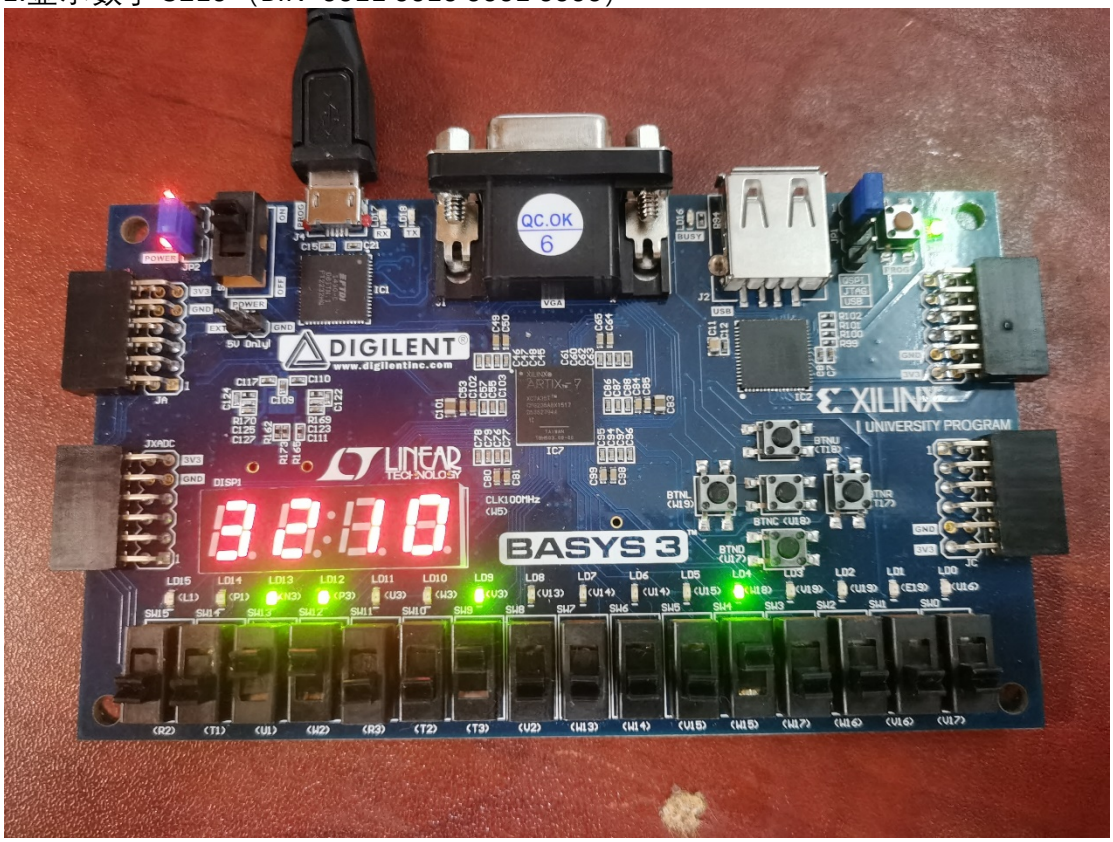
PART I（固定显示）

固定显示：2022

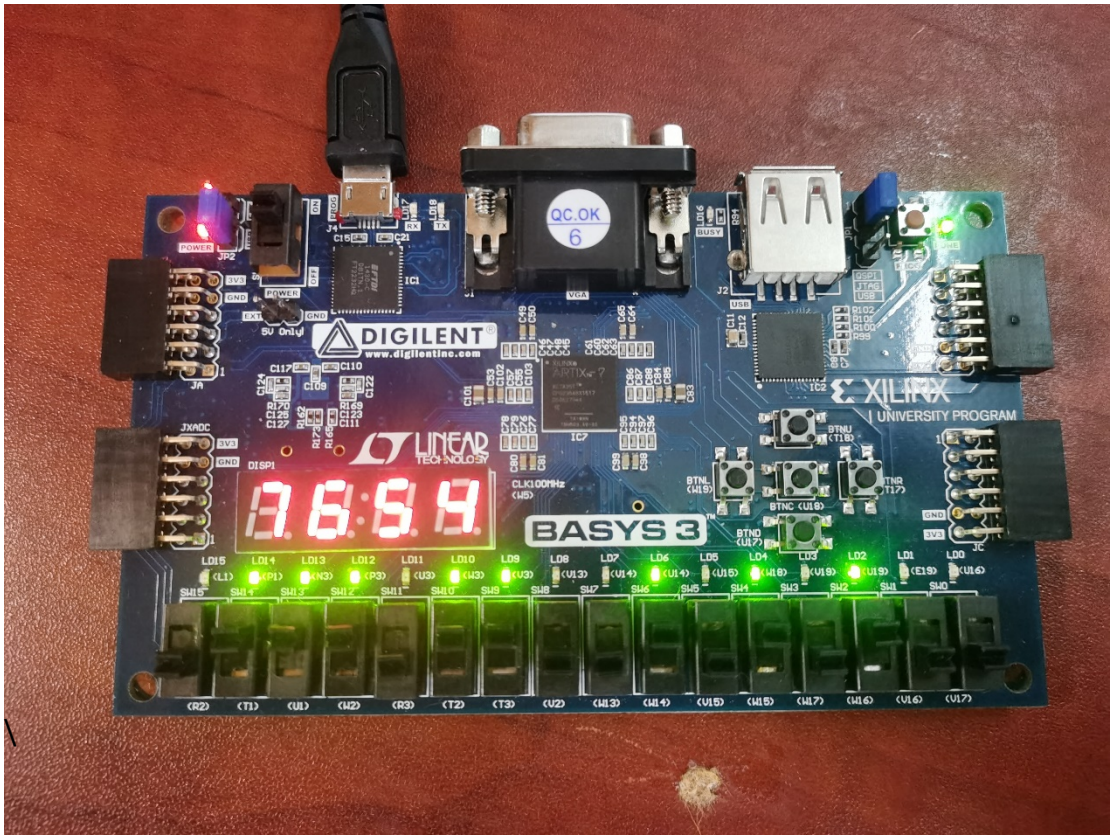


PARTII（扩展编程）

1.显示数字 3210（BIN=0011 0010 0001 0000）

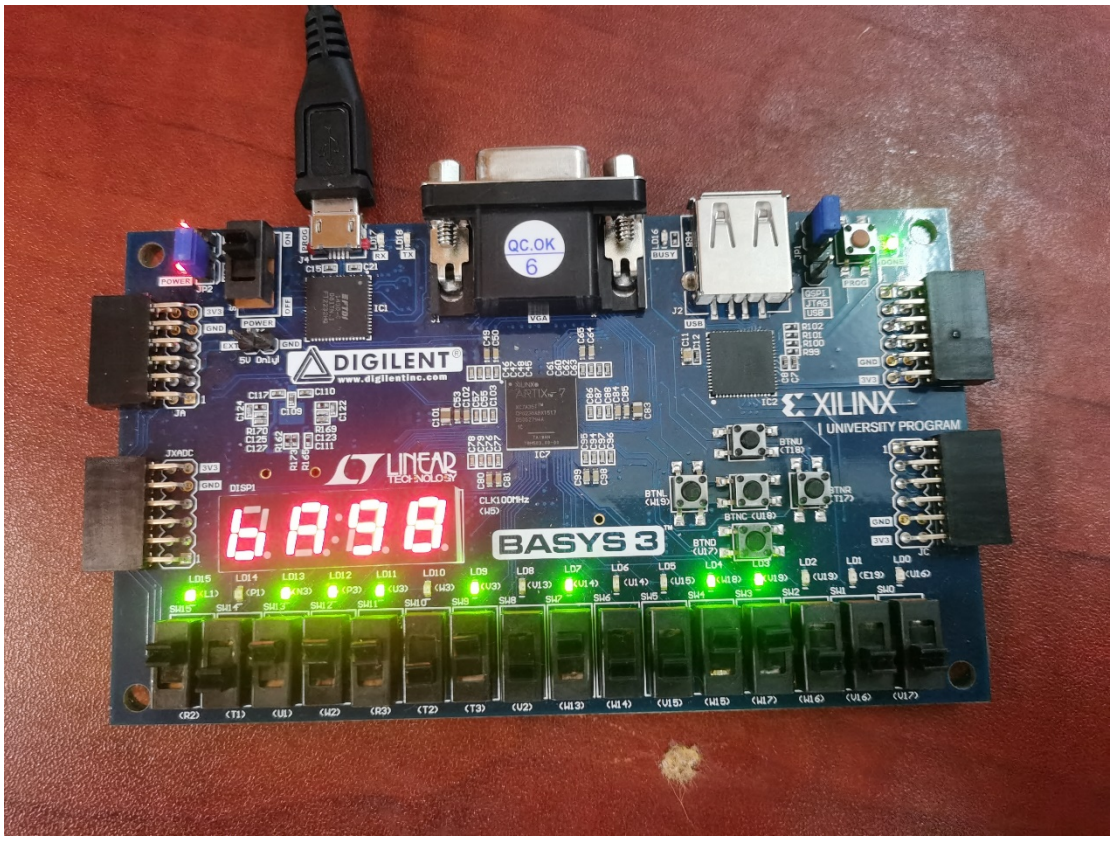


3.显示数字 7654（BIN=0111 0110 0101 0100）

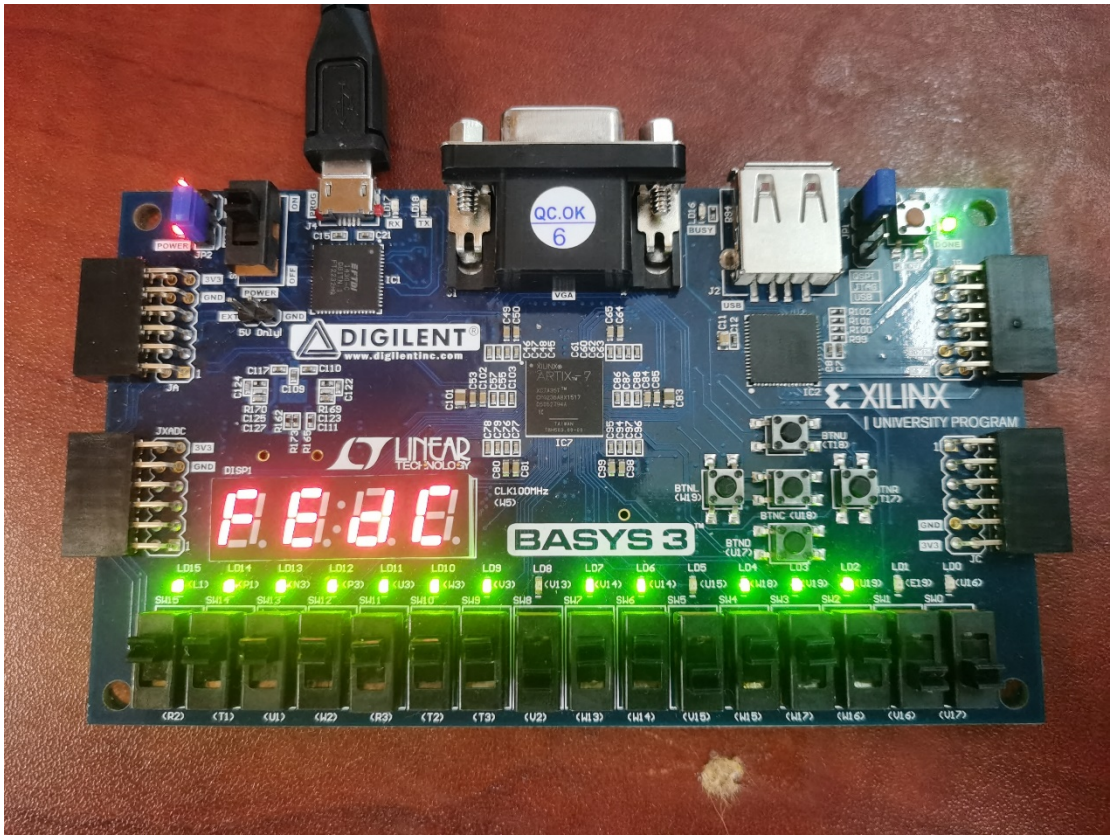


\

4.显示数字+字母组合 BA98（BIN=1011 1010 1001 1000）



5.显示字母 FEDC（BIN=1111 1110 1101 1100）



五、总结与思考

1.最初始时 Dec7Seg.sv 中并没有写入十六进制字母的数码管显示，所以需要对着图手动写，如果不写入的话默认输出“0”

2.拍照时将快门速度拉到 1/200 及以上时可以清楚观察到数码管的闪烁

3.需要注意共阳极数码管的特点是高电平熄灭低电平点亮

4.需要时刻关注数值使用的进制是否一致或者不同，否则会产生问题

5.顶层文件在关联时需要注意文件名书写正确

6.收获：掌握了用分时复用技术驱动开发板上的七段数码管