

数字电子技术 实验报告

【实验 03-ALU】

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 10 月 8 日

一、实验内容

1. 设计一个 4 位算术逻辑单元，输入信号为：两组 4 位数据输入信号（A3—A0，B3—B0），一个进位输入信号 Cin；数据输出信号为：4 位数据信号（F3—F0），一个进位输出信号 Cout。以上数据均为无符号正整数。功能控制信号有：S1、S0、M。当 M=0 时为逻辑运算，M=1 时为算术运算，S1、S0 的功能如表 1 所示。

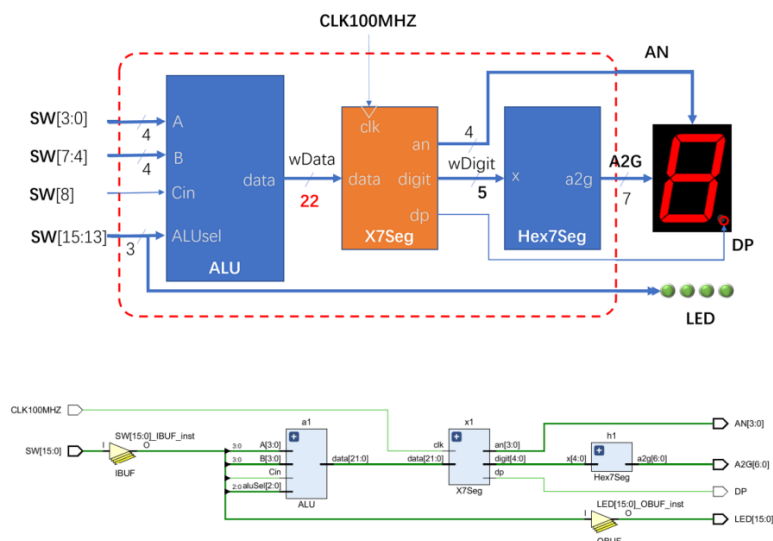
表 1. 算术逻辑单元的功能

S ₁	S ₀	M = 0 逻辑运算	M = 1 算术运算	
			C _{in} = 0	C _{in} = 1
0	0	$F = \text{not } A$	$F = A + B + 0$	$F = A + B + 1$
0	1	$F = A \text{ and } B$	$F = A - B - 0$ ($A > B$)	$F = A - B - 1$ ($A > B$)
1	0	$F = A \text{ or } B$		
1	1	$F = A \text{ xor } B$		

2. 在 BASYS3 开发板上实现上述设计，SW 选择可以自己确定。当 SW 为 1 时，其上面的 LED 点亮，否则熄灭。开发板上的 4 个七段数码管用于显示十六进制的输入数据和输出数据。

二、实验方案

参考框图



整体思路：

→输入（SW 拨码开关）

→ALU 运算模块（逻辑/算术运算+运算结果整合）

→输出模块（七段数码管分时复用+数字转七段数码管）

→输出（七段数码管+LED 灯）

三、实验分析

1. 输入 (Basys3_Master.xdc)

```
11 ## Switches
12 set_property PACKAGE_PIN V17 [get_ports {A[0]}]
13 set_property IOSTANDARD LVCMOS33 [get_ports {A[0]}]
14 set_property PACKAGE_PIN V16 [get_ports {A[1]}]
15 set_property IOSTANDARD LVCMOS33 [get_ports {A[1]}]
16 set_property PACKAGE_PIN W16 [get_ports {A[2]}]
17 set_property IOSTANDARD LVCMOS33 [get_ports {A[2]}]
18 set_property PACKAGE_PIN W17 [get_ports {A[3]}]
19 set_property IOSTANDARD LVCMOS33 [get_ports {A[3]}]
20 set_property PACKAGE_PIN W15 [get_ports {B[0]}]
21 set_property IOSTANDARD LVCMOS33 [get_ports {B[0]}]
22 set_property PACKAGE_PIN V15 [get_ports {B[1]}]
23 set_property IOSTANDARD LVCMOS33 [get_ports {B[1]}]
24 set_property PACKAGE_PIN W14 [get_ports {B[2]}]
25 set_property IOSTANDARD LVCMOS33 [get_ports {B[2]}]
26 set_property PACKAGE_PIN W13 [get_ports {B[3]}]
27 set_property IOSTANDARD LVCMOS33 [get_ports {B[3]}]
28 set_property PACKAGE_PIN W2 [get_ports {Cin}]
29 set_property IOSTANDARD LVCMOS33 [get_ports {Cin}]
30 set_property PACKAGE_PIN U1 [get_ports {S[0]}]
31 set_property IOSTANDARD LVCMOS33 [get_ports {S[0]}]
32 set_property PACKAGE_PIN T1 [get_ports {S[1]}]
33 set_property IOSTANDARD LVCMOS33 [get_ports {S[1]}]
34 set_property PACKAGE_PIN R2 [get_ports {M}]
35 set_property IOSTANDARD LVCMOS33 [get_ports {M}]
```

【分析】

开关分配：

SW[3:0]	<-	A[3:0]
SW[7:4]	<-	B[3:0]
SW[14:13]	<-	S[1:0]
SW [15]	<-	M
SW [12]	<-	Cin

2. ALU 运算模块 (mainDesign.sv)

(1) 变量定义部分

```
13 module mainDesign(
14     input logic [1:0] S, //运算功能控制器
15     input logic [3:0] A, //A输入
16     input logic [3:0] B, //B输入
17     input logic M, //逻辑or算术运算选择器
18     input logic cin, //进位输入信号
19     input logic CLK100MHZ, //时钟频率
20     output logic [1:0] SL, //SW开关LED输出,下同
21     output logic [3:0] AL,
22     output logic [3:0] BL,
23     output logic ML,
24     output logic CinL,
25     output logic [7:0] a2g, //七位数码管-阴极
26     output logic [3:0] AN //七位数码管-阳极
27 );
28
29 logic [4:0] temp; //临时运算结果储存
30 logic nf; //负数标志
31 logic zf; //零标志
32 logic cf; //进位标志
33 logic ovf; //溢出标志
34 logic Cout; //进位输出信号
35 logic [3:0] F; //运算结果返回值
36 logic [15:0] x; //模块返回值
```

【分析】 变量含义见注释

(2)ALU 模块

```
38 //ALU模块-运算部分
39 always_comb begin
40     cf=0;
41     ovf=0;
42     temp=5'b00000;
43     case(M)
44     1'b0: begin //M=0 逻辑运算
45         case(S)
46             2'b00: F= ~A; //F= not A
47             2'b01: F=A&B; //F= A and B
48             2'b10: F=A|B; //F= A or B
49             2'b11: F=A^B; //F= A xor B
50             default: F=A;
51         endcase
52     end
53     1'b1: begin //M=1 算术运算
54         case(Cin)
55             1'b0: begin //Cin=0
56                 case(S)
57                     2'b00: begin //F=A+B+0
58                         temp={1'b0,A}+{1'b0,B}+{1'b0,1'b0};
59                         F=temp[3:0];
60                         cf=temp[4];
61                         ovf=F[3]^A[3]^B[3]^cf;
62                     end
63                     2'b01: begin //F=A-B-0
64                         temp={1'b0,A}-{1'b0,B}-{1'b0,1'b0};
65                         F=temp[3:0];
66                         cf=temp[4];
67                         ovf=F[3]^A[3]^B[3]^cf;
68                     end
69                 endcase
70             end
71             1'b1: begin //Cin=1
72                 case(S)
73                     2'b00: begin //F=A+B+1
74                         temp={1'b0,A}+{1'b0,B}+{1'b0,1'b1};
75                         F=temp[3:0];
76                         cf=temp[4];
77                         ovf=F[3]^A[3]^B[3]^cf;
78                     end
79                     2'b01: begin //F=A-B-1
80                         temp={1'b0,A}-{1'b0,B}-{1'b0,1'b1};
81                         F=temp[3:0];
82                         cf=temp[4];
83                         ovf=F[3]^A[3]^B[3]^cf;
84                     end
85                 endcase
86             end
87         endcase
88     end
89 endcase
90 nf=F[3];
91 if(F==4'b0000) zf=1;
92 else zf=0;
93 end
```

【分析】

1. 运算规则基于实验内容建立
2. ALU 整体通过 always 语句实现
3. 分支结构通过 case 语句实现
4. 参考项目: ppt 中的示例工程以及 GitHub 项目:

[jrmoulton/Simple-ALU: A calculator in Verilog for the Basys3 FPGA \(github.com\)](https://github.com/jrmoulton/Simple-ALU)

(3)LED 模块&返回模块

```
95 //LED模块-开关信号控制LED灯点亮
96 assign AL=A;
97 assign BL=B;
98 assign SL=S;
99 assign ML=M;
100 assign Cin=Cin;
101
102 //返回模块-运算结果返回给输出模块
103 assign x[15:12]=A;
104 assign x[11:8]=B;
105 assign x[3:0]=F;
106 assign Cout=cf;
107 assistDesign X7(.x(x),.Cin(Cin),.clk(CLK100MHZ),.a2g(a2g),.AN(AN),.Cout(Cout));
108 endmodule
```

【分析】

1. LED-SW 的分配通过 assign 语句完成
2. A,B,F 整合成 x 返回给输出模块进行输出转换，整合通过 assign 语句完成
3. 进位输入信号，进位输出信号也返回给输出模块转换为小数点输出

3.输出模块

(1) 七段数码管分时复用 (assistDesign.sv)

```
3 module assistDesign(
4     input logic [15:0] x,
5     input logic Cin,
6     input logic Cout,
7     input logic clk,
8     input logic clr,
9     output logic [7:0] a2g,
10    output logic [3:0] AN    //数码管使能
11);
12
13 logic [1:0] s;             //选择哪个数码管
14 logic [7:0] digit;
15 logic [19:0] clkdiv;
16
17 assign s = clkdiv[19:18];   // count every 10.4ms
18
19 //4个数码管 4选1 (MUX44)
20 always_comb
21 case(s)
22 0: begin
23     if(Cout==0) digit=x[3:0]; //分配给F
24     else digit=x[3:0]+'h10; //进位标志
25 end
26 1: digit='hff;             //显示等号
27 2: begin
28     if(Cin==0) digit=x[11:8]; //分配给B
29     else digit=x[11:8]+'h10; //进位标志
30 end
31 3: digit=x[15:12];         //分配给A
32 default: digit=x[3:0];
33 endcase
34
35 //4个数码管轮流点亮
36 always_comb
37 case(s)
38 0: AN=4'b1110;
39 1: AN=4'b1101;
40 2: AN=4'b1011;
41 3: AN=4'b0111;
42 default: AN=4'b1110;
43 endcase
44
45 //时钟分频器 (20位二进制计数器)
46 always @(posedge clk, posedge clr)
47 if(clr == 1) clkdiv <= 0;
48 else      clkdiv <= clkdiv + 1;
49
50 //实例化 7段数码管
51 Dec7Seg s7(.x(digit),.a2g(a2g));
52 endmodule
```

【分析】整体参考实验 2 相关，详见注释

(2) 数字转七段数码管 (Dec7Seg.sv)

```
3 module Dec7Seg(  
4     input logic [7:0] x,  
5     output logic [7:0] a2g  
6 );  
7  
8     assign AN = 4'b0000;  
9  
10    always_comb  
11    case (x)  
12        'h00: a2g = 8'b00000011; //数字并且不显示小数点, 下同  
13        'h01: a2g = 8'b10011111;  
14        'h02: a2g = 8'b00100101;  
15        'h03: a2g = 8'b00001101;  
16        'h04: a2g = 8'b10011001;  
17        'h05: a2g = 8'b01001001;  
18        'h06: a2g = 8'b01000001;  
19        'h07: a2g = 8'b00011111;  
20        'h08: a2g = 8'b00000001;  
21        'h09: a2g = 8'b00001001;  
22        'h0a: a2g = 8'b00010001;  
23        'h0b: a2g = 8'b11000001;  
24        'h0c: a2g = 8'b01100011;  
25        'h0d: a2g = 8'b10000101;  
26        'h0e: a2g = 8'b01100001;  
27        'h0f: a2g = 8'b01110001;  
28        'h10: a2g = 8'b00000010; //数字并且显示小数点, 下同  
29        'h11: a2g = 8'b10011110;  
30        'h12: a2g = 8'b00100100;  
31        'h13: a2g = 8'b00001100;  
32        'h14: a2g = 8'b10011000;  
33        'h15: a2g = 8'b01001000;  
34        'h16: a2g = 8'b01000000;  
35        'h17: a2g = 8'b00011110;  
36        'h18: a2g = 8'b00000000;  
37        'h19: a2g = 8'b00001000;  
38        'h1a: a2g = 8'b00010000;  
39        'h1b: a2g = 8'b11000000;  
40        'h1c: a2g = 8'b01100010;  
41        'h1d: a2g = 8'b10000100;  
42        'h1e: a2g = 8'b01100000;  
43        'h1f: a2g = 8'b01110000;  
44        'hff: a2g = 8'b11101101; //显示等号  
45        default: a2g = 8'b00000000; //完全显示  
46    endcase  
47 endmodule
```

【分析】

1. 整体参考实验 2 相关和 GitHub 项目：
[nganinho/Basys3_7segments \(github.com\)](https://github.com/nganinho/Basys3_7segments)
2. 由于需要控制单个小数点，于是取消了小数点位由 DP 变量同一控制，而将其整合入了 a2g 数组和其余数码管一同控制，依旧是高电平熄灭低电平点亮
3. 等号的显示由特殊数值'hff 控制

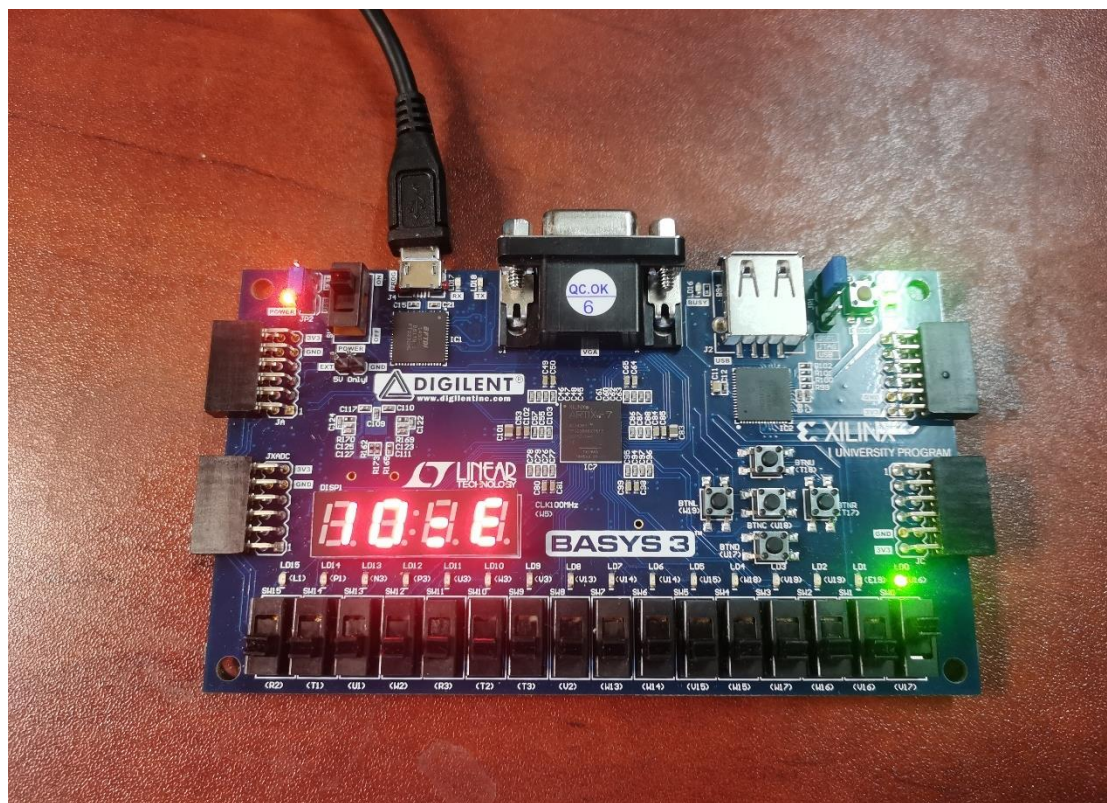
4. 输出 (Basys3_Master.xdc)

```
38 ## LEDs  
39 set_property PACKAGE_PIN U16 [get_ports {AL[0]}]  
40 set_property IOSTANDARD LVCMOS33 [get_ports {AL[0]}]  
41 set_property PACKAGE_PIN E19 [get_ports {AL[1]}]  
42 set_property IOSTANDARD LVCMOS33 [get_ports {AL[1]}]  
43 set_property PACKAGE_PIN U19 [get_ports {AL[2]}]  
44 set_property IOSTANDARD LVCMOS33 [get_ports {AL[2]}]  
45 set_property PACKAGE_PIN V19 [get_ports {AL[3]}]  
46 set_property IOSTANDARD LVCMOS33 [get_ports {AL[3]}]  
47 set_property PACKAGE_PIN W18 [get_ports {BL[0]}]  
48 set_property IOSTANDARD LVCMOS33 [get_ports {BL[0]}]  
49 set_property PACKAGE_PIN U15 [get_ports {BL[1]}]  
50 set_property IOSTANDARD LVCMOS33 [get_ports {BL[1]}]  
51 set_property PACKAGE_PIN U14 [get_ports {BL[2]}]  
52 set_property IOSTANDARD LVCMOS33 [get_ports {BL[2]}]  
53 set_property PACKAGE_PIN V14 [get_ports {BL[3]}]  
54 set_property IOSTANDARD LVCMOS33 [get_ports {BL[3]}]  
55 set_property PACKAGE_PIN P3 [get_ports {CinL}]  
56 set_property IOSTANDARD LVCMOS33 [get_ports {CinL}]  
57 set_property PACKAGE_PIN N3 [get_ports {SL[0]}]  
58 set_property IOSTANDARD LVCMOS33 [get_ports {SL[0]}]  
59 set_property PACKAGE_PIN P1 [get_ports {SL[1]}]  
60 set_property IOSTANDARD LVCMOS33 [get_ports {SL[1]}]  
61 set_property PACKAGE_PIN L1 [get_ports {ML}]  
62 set_property IOSTANDARD LVCMOS33 [get_ports {ML}]  
65 ##7 segment display  
66 set_property PACKAGE_PIN W7 [get_ports {a2g[7]}]  
67 set_property IOSTANDARD LVCMOS33 [get_ports {a2g[7]}]  
68 set_property PACKAGE_PIN W6 [get_ports {a2g[6]}]  
69 set_property IOSTANDARD LVCMOS33 [get_ports {a2g[6]}]  
70 set_property PACKAGE_PIN U8 [get_ports {a2g[5]}]  
71 set_property IOSTANDARD LVCMOS33 [get_ports {a2g[5]}]  
72 set_property PACKAGE_PIN V8 [get_ports {a2g[4]}]  
73 set_property IOSTANDARD LVCMOS33 [get_ports {a2g[4]}]  
74 set_property PACKAGE_PIN U5 [get_ports {a2g[3]}]  
75 set_property IOSTANDARD LVCMOS33 [get_ports {a2g[3]}]  
76 set_property PACKAGE_PIN V5 [get_ports {a2g[2]}]  
77 set_property IOSTANDARD LVCMOS33 [get_ports {a2g[2]}]  
78 set_property PACKAGE_PIN U7 [get_ports {a2g[1]}]  
79 set_property IOSTANDARD LVCMOS33 [get_ports {a2g[1]}]  
80 set_property PACKAGE_PIN V7 [get_ports {a2g[0]}]  
81 set_property IOSTANDARD LVCMOS33 [get_ports {a2g[0]}]  
82  
83 set_property PACKAGE_PIN U2 [get_ports {AN[0]}]  
84 set_property IOSTANDARD LVCMOS33 [get_ports {AN[0]}]  
85 set_property PACKAGE_PIN U4 [get_ports {AN[1]}]  
86 set_property IOSTANDARD LVCMOS33 [get_ports {AN[1]}]  
87 set_property PACKAGE_PIN V4 [get_ports {AN[2]}]  
88 set_property IOSTANDARD LVCMOS33 [get_ports {AN[2]}]  
89 set_property PACKAGE_PIN W4 [get_ports {AN[3]}]  
90 set_property IOSTANDARD LVCMOS33 [get_ports {AN[3]}]
```

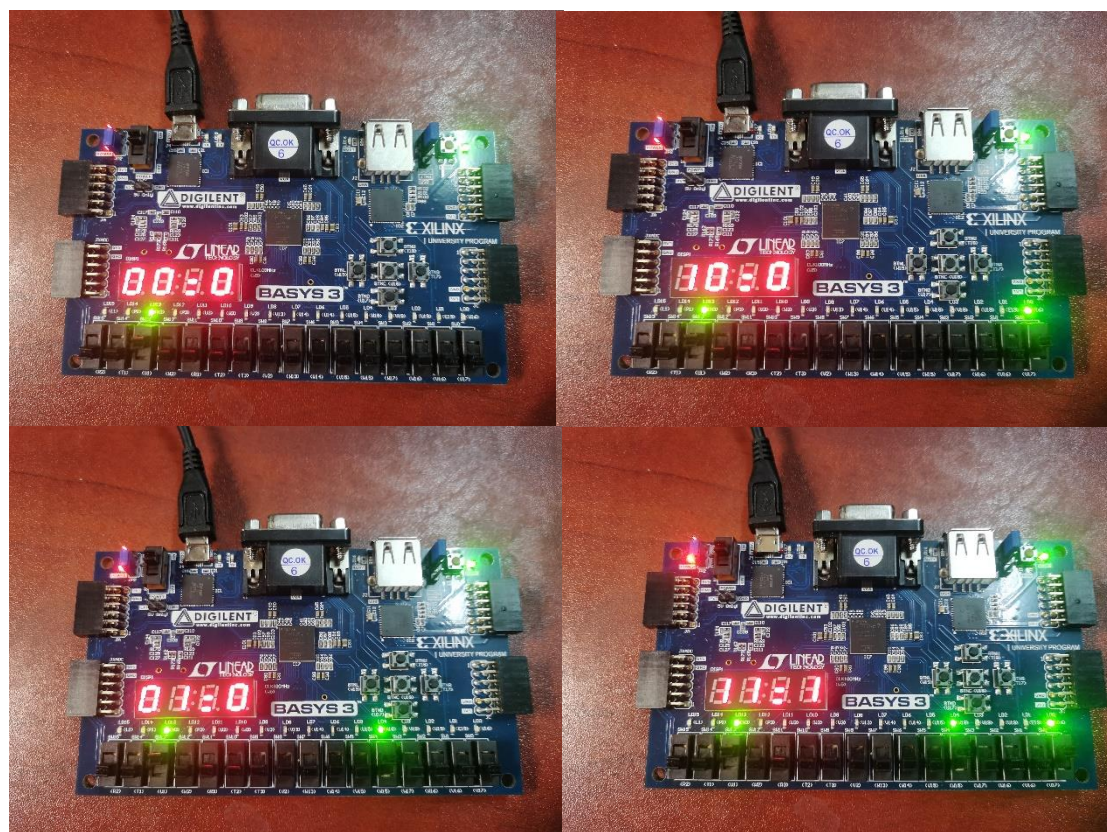

四、实验结果

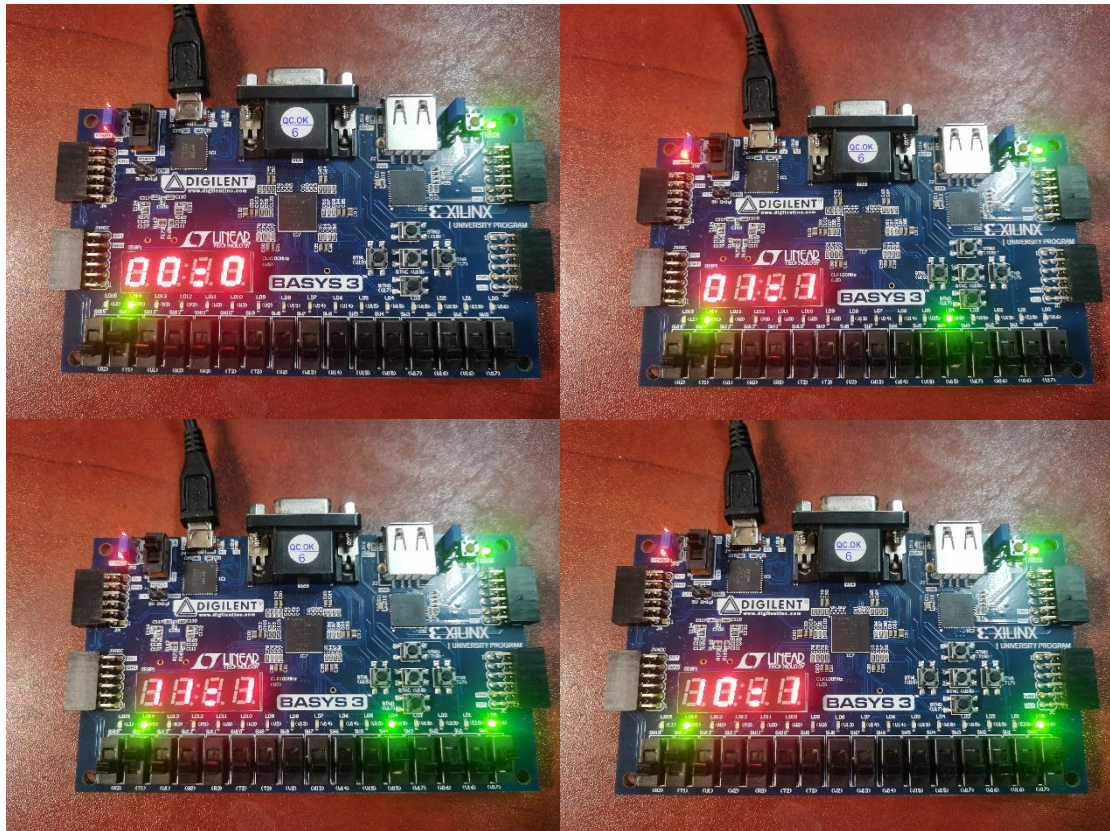
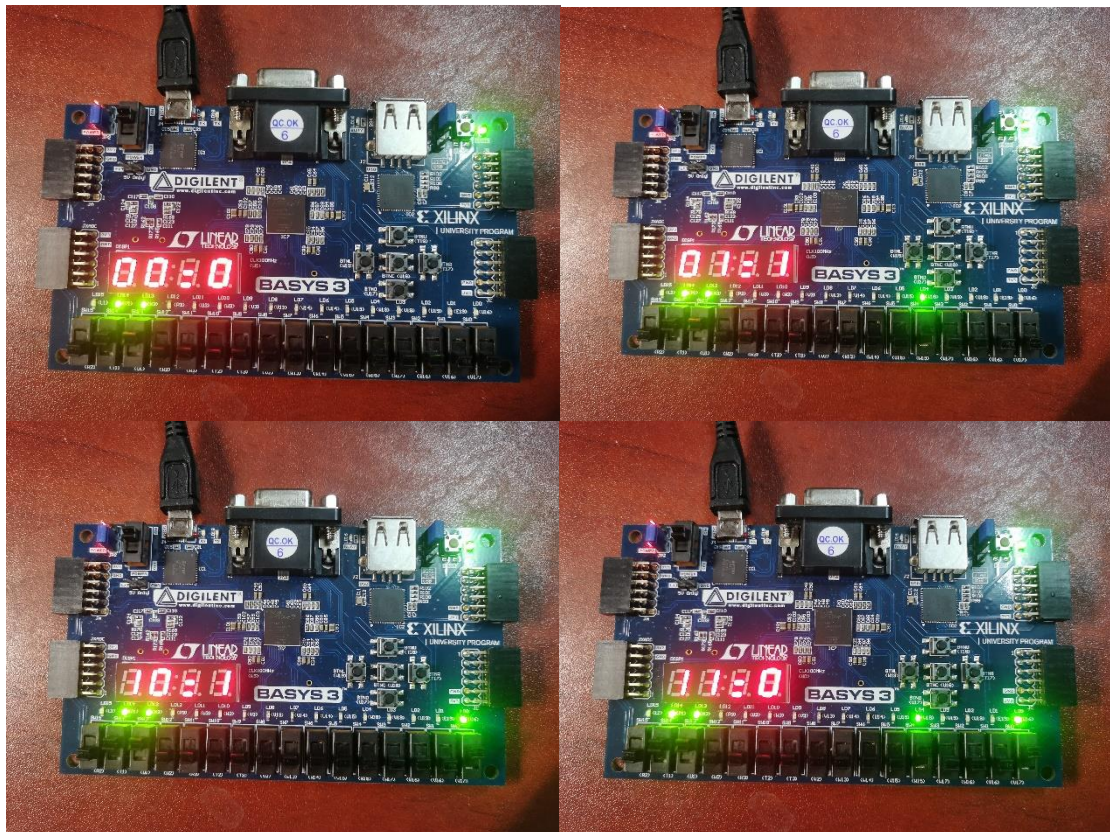
1.M=0 逻辑运算

(1) 非门 ($S1=0$ $S0=0$)

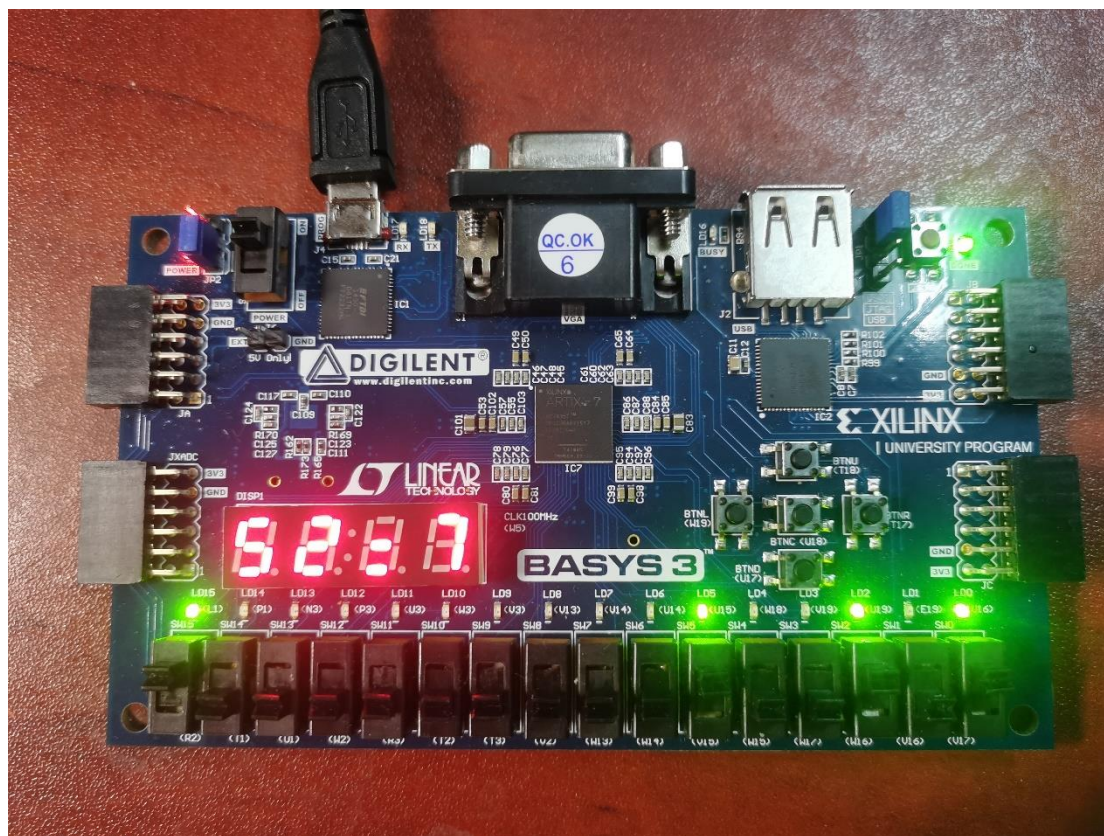
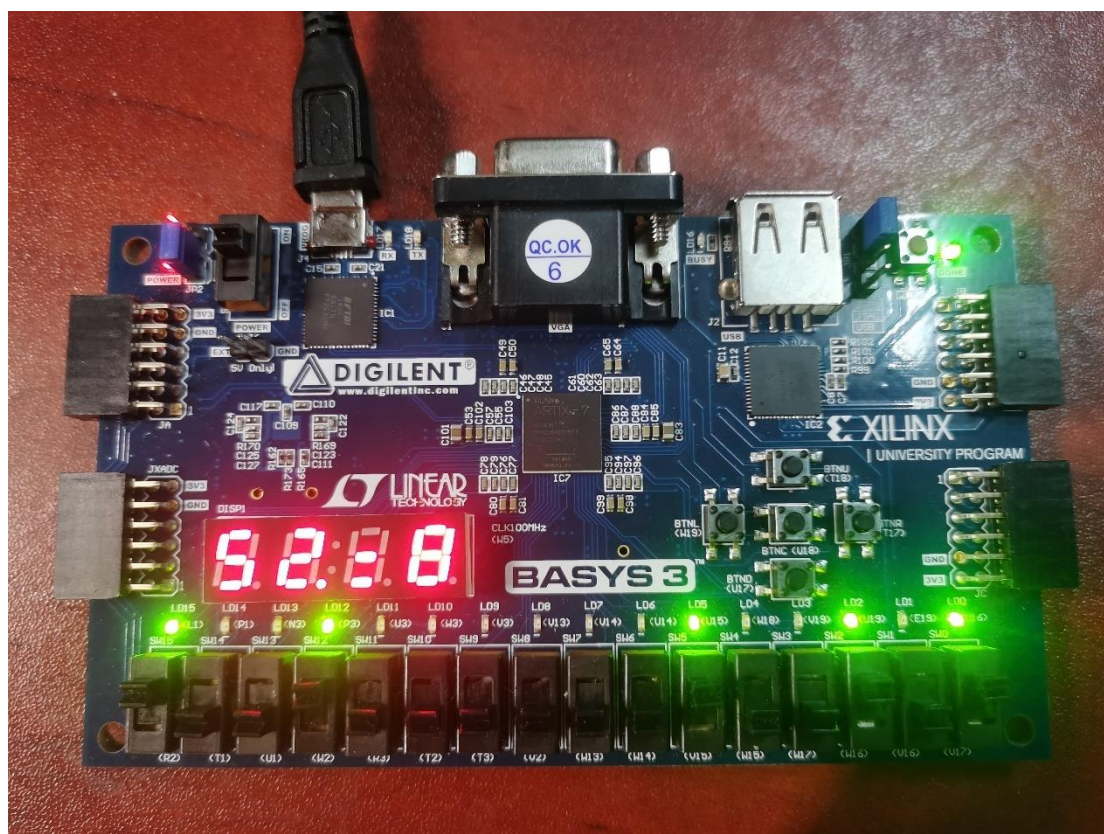


(2) 与门 ($S1=0$ $S0=1$)

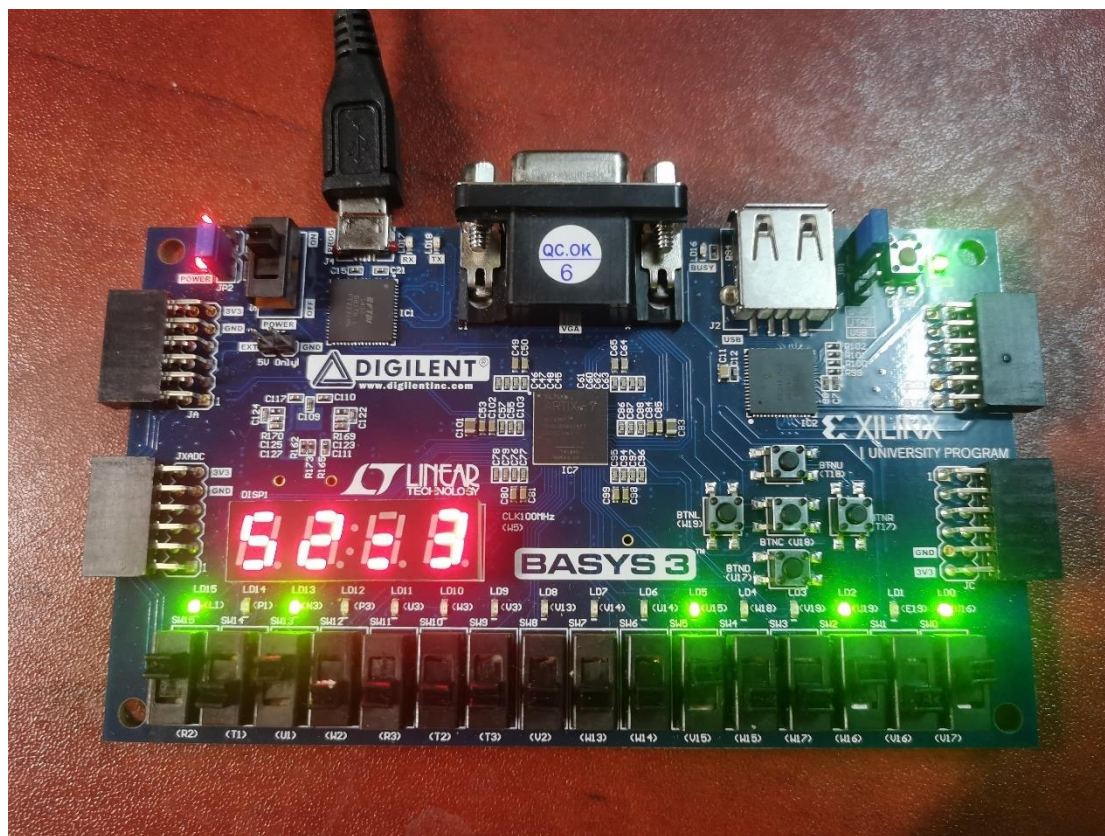


(3) 或门 ($S1=1$ $S0=0$)(4) 异或门 ($S1=1$ $S0=1$)

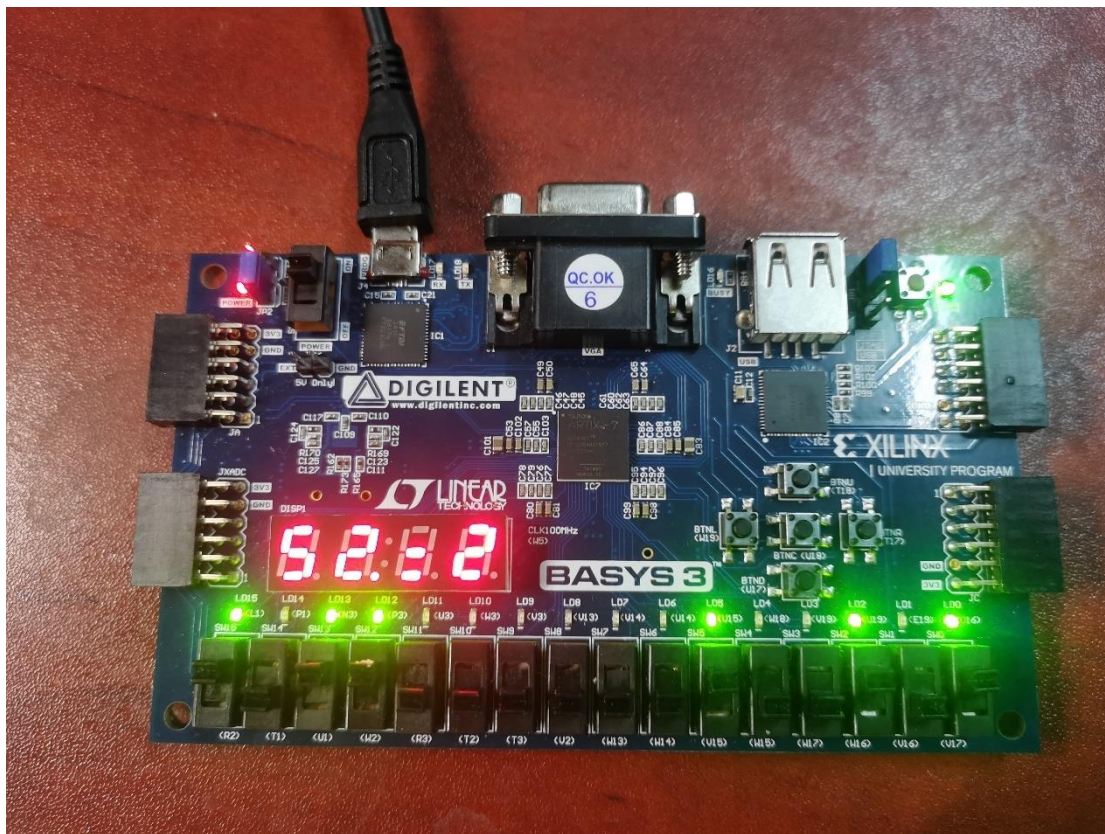
2.M=1 算术运算

(1) 加法 ($S1=0$ $S0=0$ $Cin=0$)(2) 加法 ($S1=0$ $S0=0$ $Cin=1$)

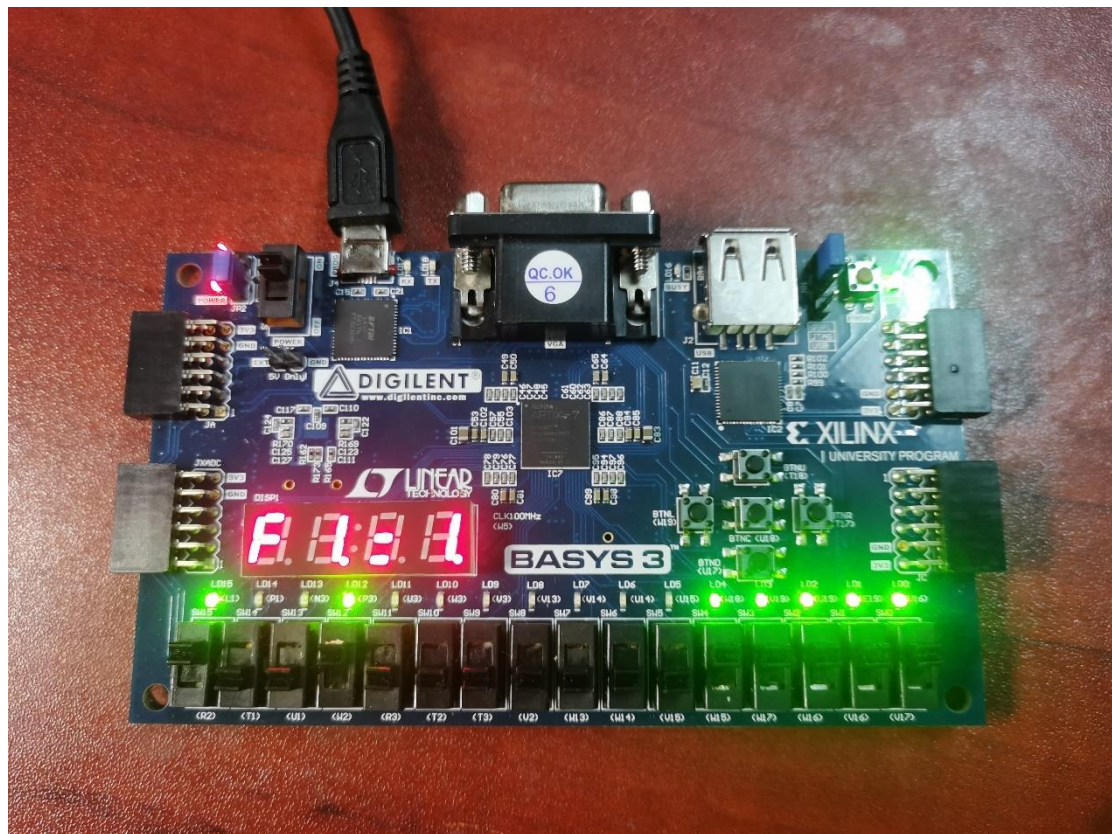
(3) 减法 ($S1=0$ $S0=0$ $Cin=0$)



(4) 减法 ($S1=0$ $S0=0$ $Cin=1$)



(5) Cout=1



五、总结与思考

- 1.在 ALU 中定义的输入输出变量需要和约束文件中定义的名称保持一致
- 2.在实验二中所有的小数点由 DP 变量统一控制点亮/熄灭，而本实验中则需要单独控制第二位和第四位的小数点，所以需要将原 DP 整合进入 a2g 数组统一控制，a2g 的长度从原来的 7 位二进制数变为 8 位二进制数
- 3.由于增加了小数点和等号的显示，实例化时需要返回 2 位十六进制数才能完整表示全部情况
- 4.收获：ALU 模块中 always 语句，case 语句，if 语句的使用，进行简单的逻辑与算术运算；输出模块中重写数码管约束文件和实例化文件