第一次实验实验报告

-加法器实验

陈刚老师班级 22336313(学号)-郑鸿鑫

1. 实验内容

本实验实现了一个 4 位的加法器，并且加法的结果显示到 7 段数码管上面，用一个 Led 灯来显示加法器是否有溢出。然后将上述加法器扩展位 8 位的加法器同样有显示功能（其中注意数码管只有 4 位，故采用 16 进制的方式显示到数码管上，并且为了不引起歧义和方便，某些字母采用了小写字母进行表示）。一样的，8 位加法器也具有溢出检测功能。

2. 实验过程

先在 vivado 上建立工程，编写实现代码。然后进行仿真综合，待结果无误后，连接 basys3 开发板运行程序。

8 位加法器的代码如下：（解释已经写在注释中）

系统由一个主要的模块 adder_8_实现，且该模块不但有将 2 个输入相加的功能，还有将结果转为 16 进制输出在数码管上的功能。另一个模块 ADDer_8 则是进行了对其的实例化

```
module adder_8_(//建立一个8进制加法器，并将结果以两位16进制显示在数码管上面
    input clk,//时钟信号
    input [7:0] SW1,//接收第一个8位二进制数
    input [7:0] SW2,//接收第二个8位二进制数
    output reg [10:0] seg,//控制数码管的显示
    output reg overflow = 0//判断是否溢出的信号
    );
    reg [19:0]count = 0;//用于计数时钟上升沿的次数，本质是为了降低时钟频率，详见下面代码
    reg [2:0]sel = 0;//选择控制数码管的哪一个显示，详见下面代码
    reg cas = 0;//用于选择2位16进制在数码管上显示的第一位或者第二位，详见下列代码
    parameter T1MS = 50000;//和count搭配使用，起相同作用
    reg [8:0] sum;//用于存储两个二进制数的和
```

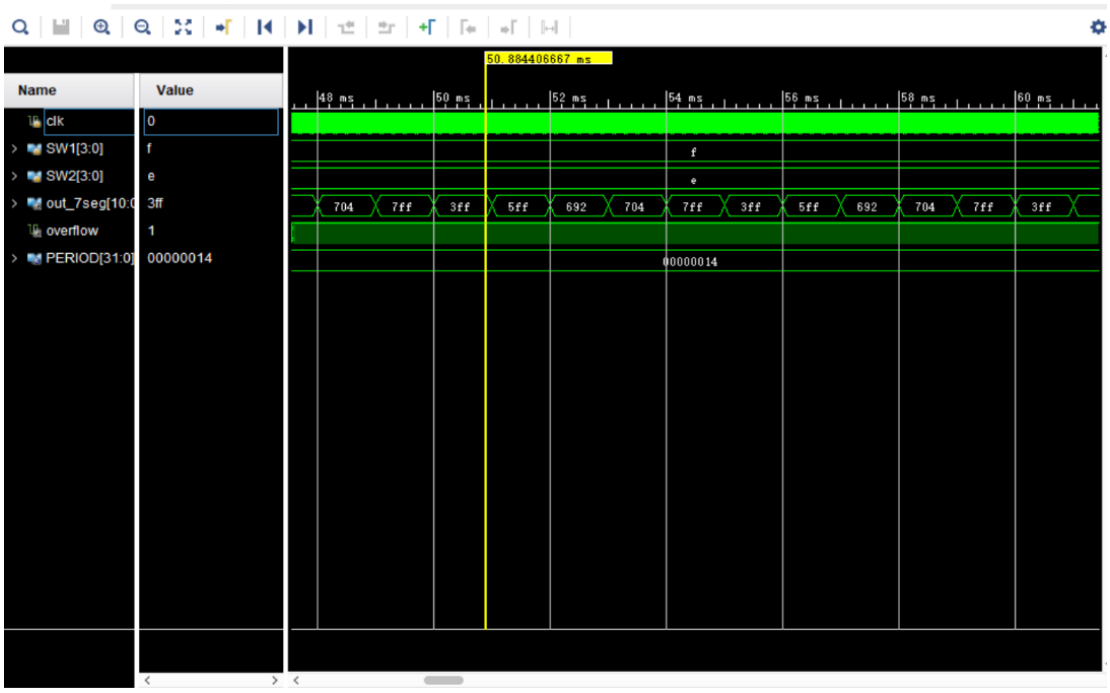
```
always@ (posedge clk)//每个时钟上升沿到来时
begin
    sum <= SW1 + SW2;//相加
    if(sum[8] == 1) overflow <= 1;//如果最高位为1，则溢出信号为1
    else overflow <= 0;//否则为0
    case(cas)
    0://cas 为0，则选择数码管的低位显示
        case(sum[3:0])//用 sum 的后 4 位来选择 0~f 的数码管显示
        0: seg <= 11'b1110_0000001;
        1: seg <= 11'b1110_1001111;
        2: seg <= 11'b1110_0010010;
        3: seg <= 11'b1110_0000110;
        4: seg <= 11'b1110_1001100;
        5: seg <= 11'b1110_0100100;
        6: seg <= 11'b1110_0100000;
        7: seg <= 11'b1110_0001111;
        8: seg <= 11'b1110_0000000;
        9: seg <= 11'b1110_0000100;
        10: seg <= 11'b1110_0001000;
        11: seg <= 11'b1110_1100000;
        12: seg <= 11'b1110_0110001;
        13: seg <= 11'b1110_1000010;
        14: seg <= 11'b1110_0010000;
        15: seg <= 11'b1110_0111000;
        default:seg <= 11'b1111_1111111;
    endcase
    1://cas 为1，则选择数码管的高位显示
        case(sum[7:4])//用 sum 的高 4 位来选择 0~f 的数码管显示
        0: seg <= 11'b1101_0000001;
        1: seg <= 11'b1101_1001111;
        2: seg <= 11'b1101_0010010;
        3: seg <= 11'b1101_0000110;
        4: seg <= 11'b1101_1001100;
        5: seg <= 11'b1101_0100100;
        6: seg <= 11'b1101_0100000;
        7: seg <= 11'b1101_0001111;
        8: seg <= 11'b1101_0000000;
        9: seg <= 11'b1101_0000100;
        10: seg <= 11'b1101_0001000;
        11: seg <= 11'b1101_1100000;
        12: seg <= 11'b1101_0110001;
        13: seg <= 11'b1101_1000010;
        14: seg <= 11'b1101_0010000;
        15: seg <= 11'b1101_0111000;
        default:seg <= 11'b1111_1111111;
        endcase
    endcase
end
always@(posedge clk)//时钟上升沿到来时
begin
    count <= count + 1;//每次时钟正边沿到来时计数
    if(count == T1MS)
    begin
        count <= 0;//count 到达最大值时重置清零
```

```
            sel <= sel + 1;//挑选数码管，以起到动态扫描的作用
            if(sel == 4) sel <= 0;
            cas <= cas + 1;//总共有2个数码管，所以每次都翻转一次，溢出的高位不管，则只需对其加1
        end
    end
endmodule
module ADDer_8;
    reg clk;
    reg [7:0] SW1;//接收第一个8位二进制数
    reg [7:0] SW2;//接收第二个8位二进制数
    wire [10:0] seg;//用于控制数码管的显示
    wire overflow;//作为溢出的信号
    adder_8_ uu(
        .clk(clk),
        .SW1(SW1),
        .SW2(SW2),
        .seg(seg),
        .overflow(overflow)
    );//实例化模块adder_8_
    initial begin
        clk = 0;//时钟信号初始化位0
        SW1 = 80;//给第一个数赋初值80
        SW2 = 240;//给第二个数赋初值240
    #100;
    end
    parameter PERIOD = 20;
    always begin
    clk=0;
    #(PERIOD/2);
    clk=1;
    #(PERIOD/2);
    end
endmodule
```

其中 4 位加法器的代码与 8 位的同理，不再放出来展示

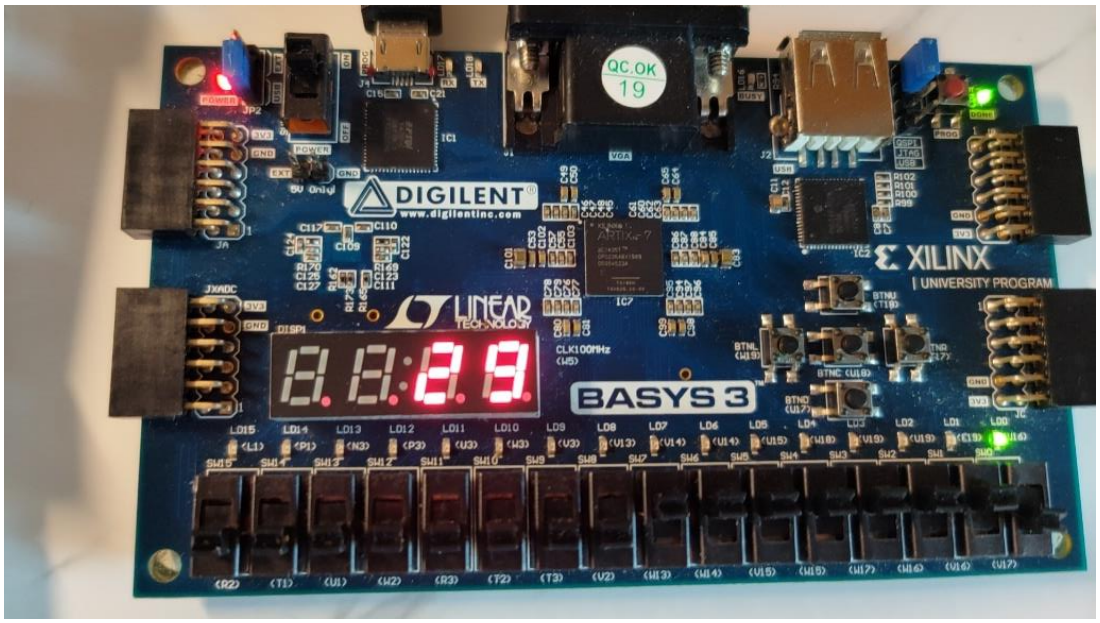
3. 实验结果分析

4 位加法器仿真界面（以输入 14 15 相加为例）



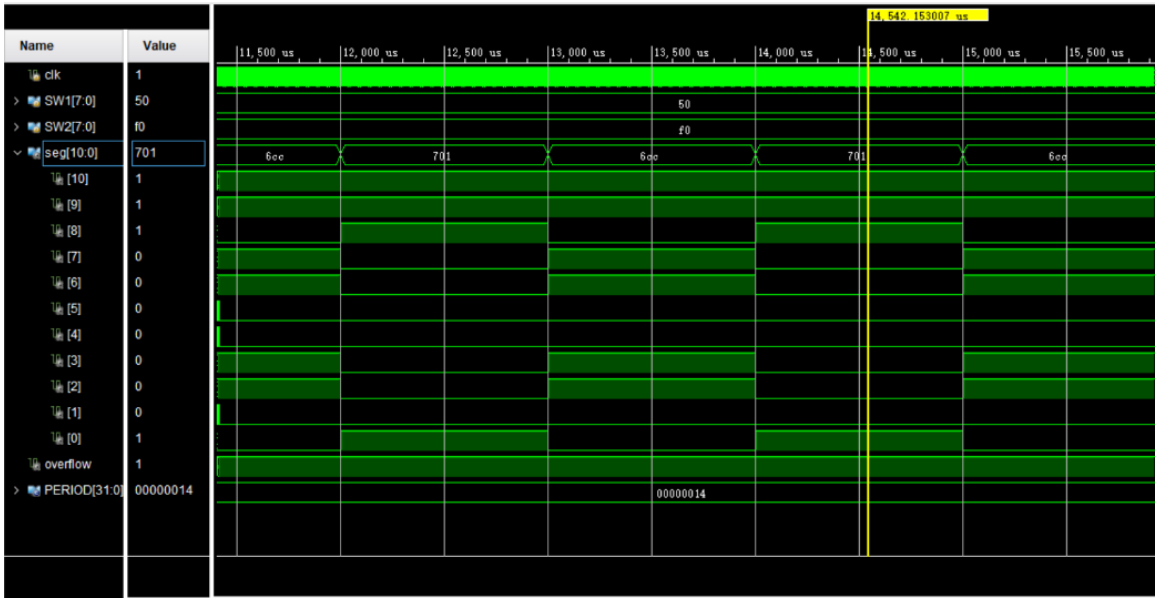
如图所示，第一个数正常被赋值为 14，第二个数正常被赋值为 15，此时溢出信号变为 1，符合预期（由于和为 29 已经超过了 15）同时观察到 out_seg7 的值会在时钟上升沿到来时改变，以控制不同的数码管显示对应的数字，由于时钟的频率非常快会产生动态扫描的效果。

4 位加法器运行实拍



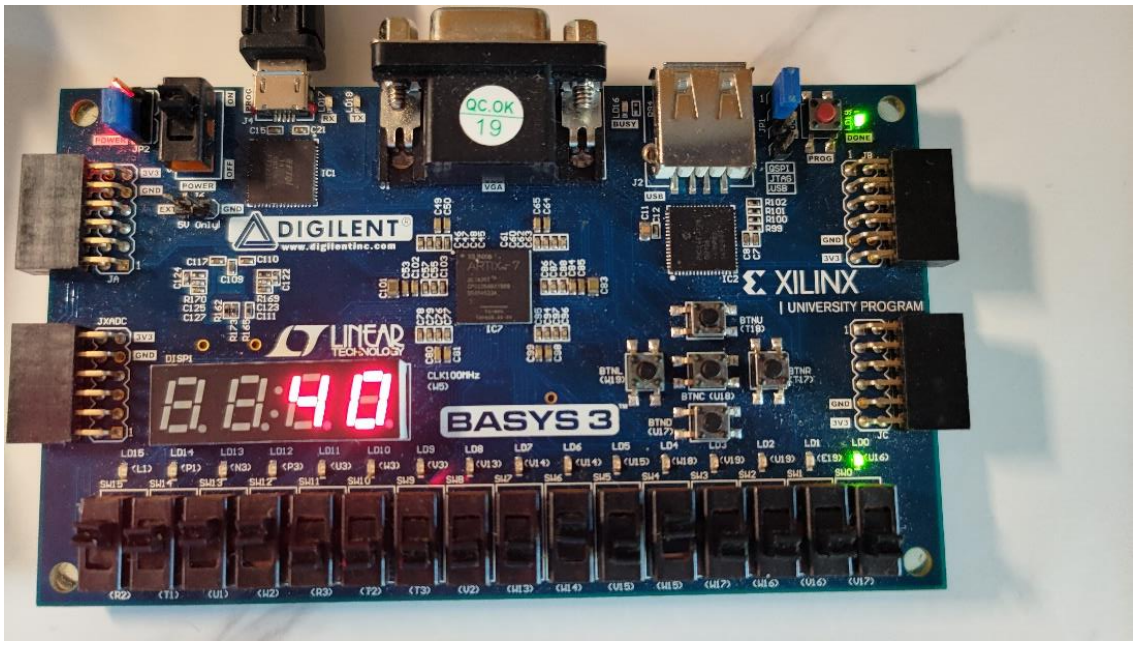
如图所示，14+15 正常显示得到 29，且溢出信号灯亮起，符合预期。

8 位加法器仿真界面（以输入 80 240 为例）



如图所示，第一个 8 位二进制数被赋初值位 80，对应 16 进制的 50，第二个 8 位二进制数被赋初值为 240，对应 16 进制的 f0，正常。此时相加的结果为 320 > 256，所以溢出信号变为 1，正常。同时，在每个时钟上升沿到来的时候 seg 会相应的改变以控制不同的数码管进行显示，由于时钟频率非常快，会产生动态扫描的效果，使 4 个数码管看起来是同时亮起。符合预期。

8 位加法器运行实拍



80 加 240 的 16 进制结果为 140，其中 1 作为溢出信号显示在信号灯上 40 作为两位低位 16 进制数显示在数码管上，与预期符合良好。

4. 实验总结

实验过程中遇到的问题：

在一个 always 语句中不能对一个变量进行两次赋值 否则会导致两次赋值都失败，然后数码管显示失效，后来通过增加一个 reg 变量来控制赋值的选择解决了问题，使数码管显示正常。

实验结果与预期符合良好，实现了 4 位以及 8 位加法器的正常功能。