组成原理课程第_二_次实验报告

实验名称：定点乘法

学号：2012679 姓名：王娇妹 班次：张金老师

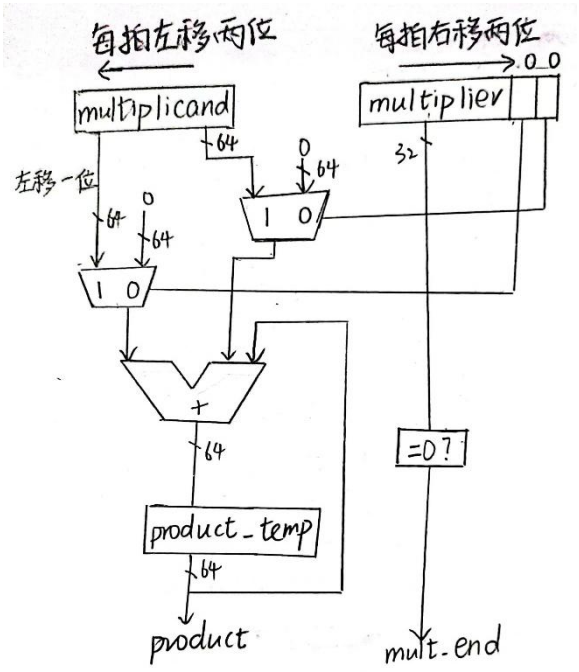
一、实验目的

1. 理解定点乘法的不同实现算法的原理，掌握基本实现算法。
2. 熟悉并运用 verilog 语言进行电路设计。
3. 为后续设计 cpu 的实验打下基础。

二、实验内容说明

1．在原有迭代乘法的基础上，实现一次移两位的迭代乘法。
2．波形仿真，并分析一位乘法和两位乘法对应的时钟周期数的差别。
3．画出两位迭代乘法的原理图。

三、实验原理图



运算时乘数每次右移两位，被乘数每次左移两位。需要定义两个临时变量 partial_product1 和 partial_product2。

partial_product1 存乘数末位与被乘数的积：乘数末位为 1，partial_product1 是被乘数本身；乘数末位为 0，partial_product1 就是 0。

partial_product2 存乘数倒数第二位与被乘数的积：乘数倒数第二位为 1，partial_product2 是被乘数左移一位；乘数倒数第二位为 0，partial_product2 就是 0。

当乘数为 0 时，结束运算。

四、实验步骤

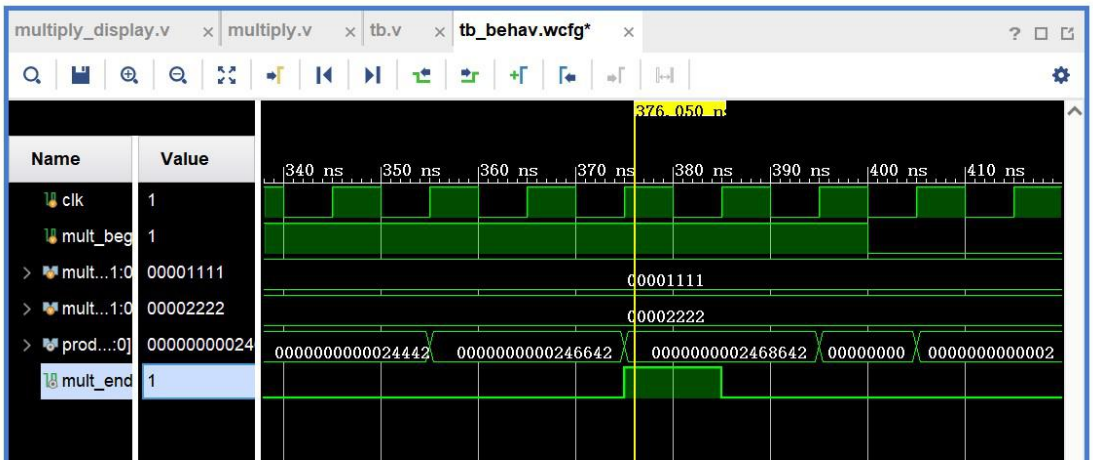
实验改进主要修改了 multiply.v 文件，运算时每次移动两位。

具体代码及分析如下：

```
reg   [63:0] multiplicand; //加载被乘数，运算时每次左移两位
always @ (posedge clk)
begin
```

```
        if (mult_valid)
        begin    // 如果正在进行乘法，则被乘数每时钟左移两位
            multiplicand <= {multiplicand[61:0],2'b00};
        end
        else if (mult_begin)
        begin    // 乘法开始，加载被乘数，为乘数 1 的绝对值
            multiplicand <= {32'd0,op1_absolute};
        end
    end
    reg   [31:0] multiplier; //加载乘数，运算时每次右移两位
    always @ (posedge clk)
    begin
        if (mult_valid)
        begin     // 如果正在进行乘法，则乘数每时钟右移两位
            multiplier <= {2'b00,multiplier[31:2]};
        end
        else if (mult_begin)
        begin     // 乘法开始，加载乘数，为乘数 2 的绝对值
            multiplier <= op2_absolute;
        end
    end
     // 部分积，定义两个临时变量
    wire [63:0] partial_product1; //存乘数末位数与被乘数的积
    wire [63:0] partial_product2;   //存乘数倒数第二位数与被乘数的积
    //乘数末位为 1，临时变量 1 是被乘数本身；乘数末位为 0，临时变量 1 就是 0
    assign partial_product1 = multiplier[0] ? multiplicand : 64'd0;
    //乘数倒数第二位为 1，临时变量 2 由被乘数左移一位得到；乘数倒数第二位为 0，
    临时变量 2 就是 0
    assign partial_product2 = multiplier[1]?{multiplicand[62:0],1'b0} : 64'd0;
    reg [63:0] product_temp; //累加器
    always @ (posedge clk)
    begin
        if (mult_valid)
        begin
            product_temp <= product_temp + partial_product1+ partial_product2;
        end
        else if (mult_begin)
        begin
            product_temp <= 64'd0;   // 乘法开始，乘积清零
        end
    end
```

五、实验结果分析



输入：

mult_op1 = 00001111

mult_op2 = 00002222

输出：

product = 2468642

1111*2222 = 2468642，实验测试结果正确。

周期数：

经过实验，发现两位乘法器的时钟周期是一位乘法器的一半。

六、总结感想

一位迭代乘法的结束标志为乘数移位后为 0，所以对于 X 位乘法，最多需要 X 拍才能完成一次乘法。而改进后的两位迭代乘法，每次移动两位数，最多需要 X/2 拍就能完成一次乘法，这提高了运算速度。