

组成原理课程第 二 次实报告

实验名称：定点乘法器改进

学号： 2213041 姓名： 李雅帆 班次： 李涛老师

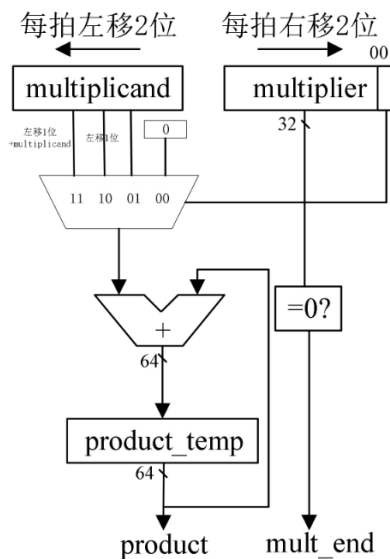
一、实验目的

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。

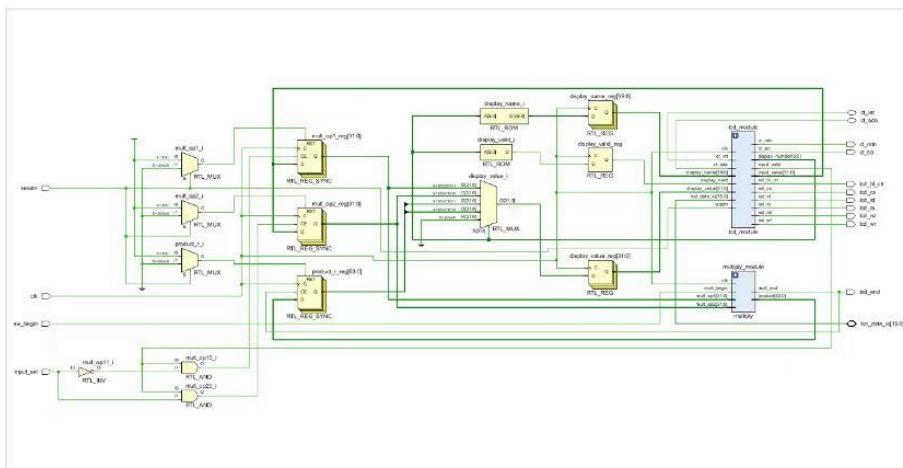
二、实验内容说明

完成乘法器的改进，将乘法器改成被乘数每次左移两位，从而减少一半的运算周期。使用波形图进行仿真，并分析一位乘法和两位乘法对应的时钟周期数的差别，最后上箱验证。

1. 实验原理图如下：



2. 实验顶层模块框图如下：



三、实验步骤

在每一拍判断当前是否还在进行乘法运算，如果还在进行乘法运算，则将被乘数左移两位，乘数对应右移两位。然后通过乘数的低两位的判断，来确定对部分积的结果做何种运算。

1. 对被乘数代码的修改

```
//加载被乘数，运算时每次左移两位
reg [63:0] multiplicand;
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
end
```

2. 对乘数代码的修改

```
//加载乘数，运算时每次右移一位
reg [31:0] multiplier;
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
end
```

3. 对部分积的代码运算的修改

```
// 部分积
wire [63:0] partial_product1;
wire [63:0] partial_product2; //定义两个临时变量
assign partial_product1 = multiplier[0] ? multiplicand : 64'd0;
//乘数末位为1，临时变量1是乘数本身；乘数末位为0，临时变量1是0
assign partial_product2 = multiplier[1] ? {multiplicand[62:0], 1'b0} : 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是0
```

4. 对累加器的修改

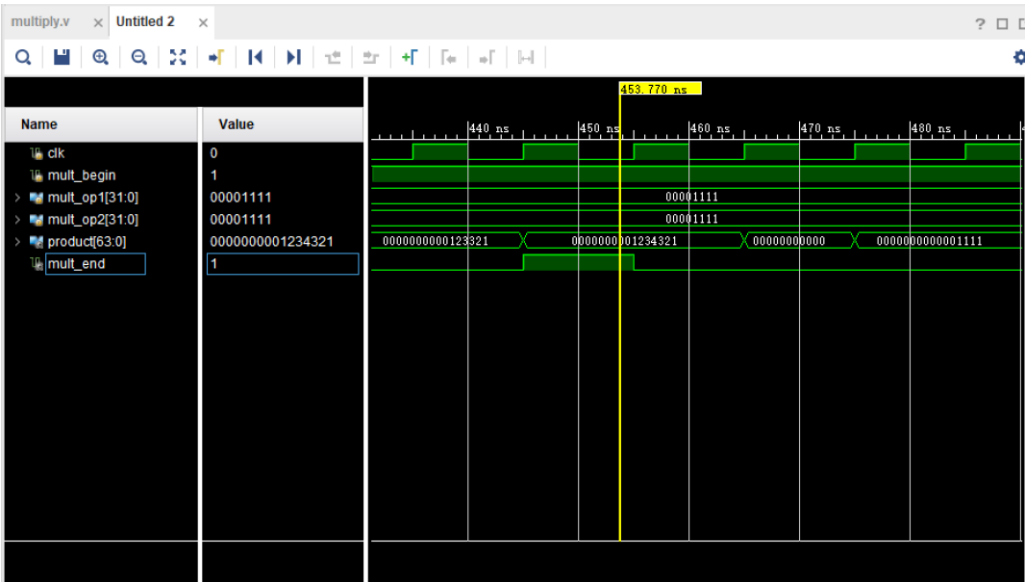
```
//累加器
reg [63:0] product_temp;
always @ (posedge clk)
begin
    if (mult_valid)
    begin
        product_temp <= product_temp + partial_product1+ partial_product2;
    end
    else if (mult_begin)
    begin
        product_temp <= 64'd0; // 乘法开始，乘积清零
    end
end
end
```

四、实验结果分析

1. 仿真结果：

由波形图可知，与原始的乘法相比，每次位移两位的方式能够明显缩短计算所需要的时钟周期数。

原始的方法完成乘法计算所需要的时钟周期数为 16，而改进后的方法完成乘法计算所需要的时钟周期数为 8，由于每次位移两位，因此相较于原始的方法，其效率为原来的两倍。



2. 上箱烧录，实验箱验证。





通过验证可知答案正确。

五、总结感想

这个实验让我更深入地理解了定点乘法的不同实现算法的原理，以及如何使用 Verilog 语言进行电路设计。

通过改进乘法器，将被乘数每次左移两位，我学会了如何优化乘法器的设计以减少运算周期，体会了通过一次性位移多位来加速乘法运算，两位乘法所需要使用的时钟周期只是一位乘法使用时钟周期的一半；知道了在进行乘法运算的过程中，可以通过复用被乘数来降低进行乘法运算时的存储开销。

在进行仿真和波形图分析的过程中，我能够清晰地看到一位乘法和两位乘法之间在时钟周期数上的差异，更好地理解乘法器的性能。