

组成原理课程第 二 次实验报告

实验名称	数据运算：定点乘法			班级	0981 张金
学生姓名	林盛森	学号	2312631	指导老师	董前琨
实验地点	实验楼 A210		实验时间	2025.3.19	

一、 实验目的

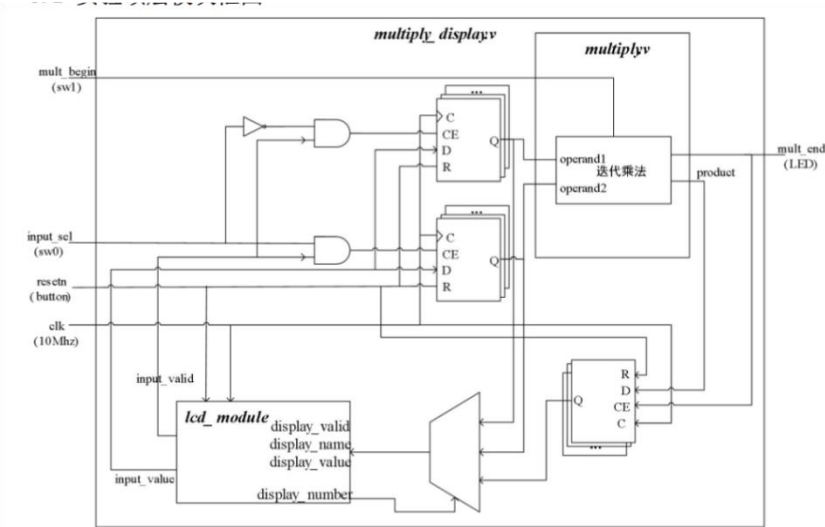
- 1. 理解定点乘法的不同实现算法的原理，掌握基本实现算法。
- 2. 熟悉并运用 verilog 语言进行电路设计。
- 3. 为后续设计 cpu 的实验打下基础。

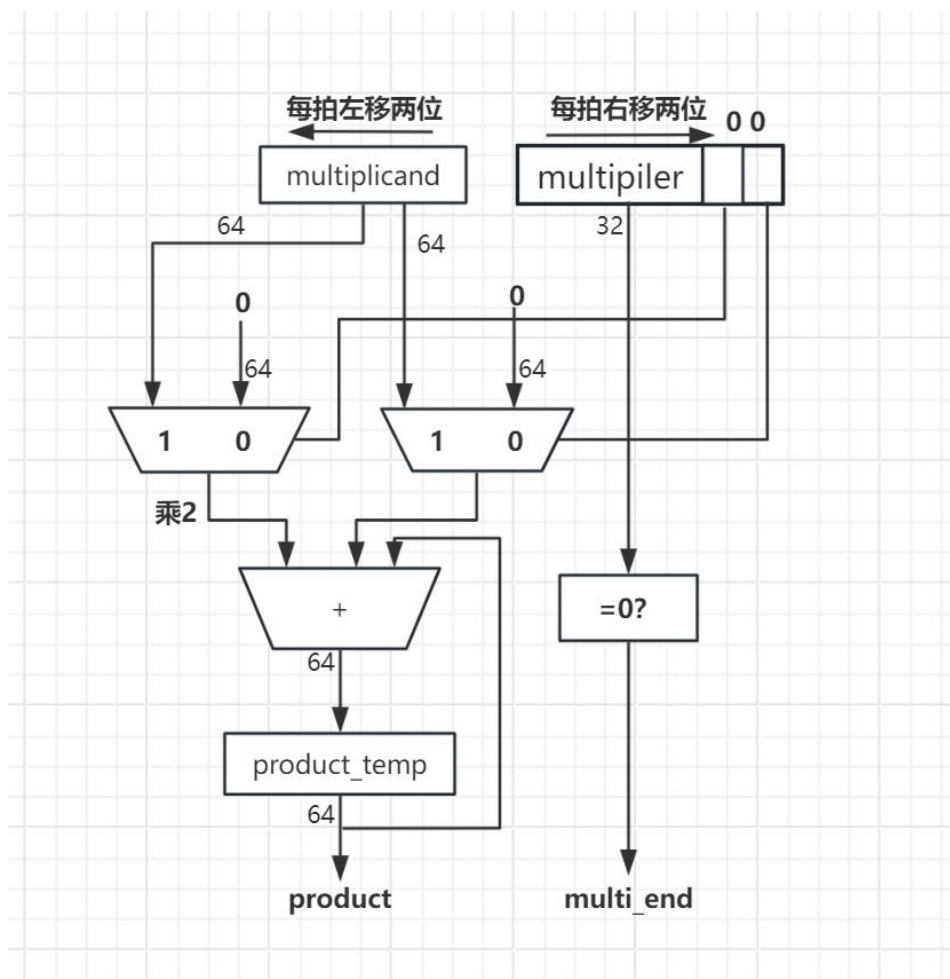
二、 实验内容说明

- 1. 参考实验手册了解两个定点数的相乘的电路设计，熟练使用 vivado、Verilog 语言的基本使用，并学会如何对电路进行实现以及实验箱的使用；
- 2. 学习并理解计算机中定点乘法器的多种实现算法的原理，重点掌握迭代乘法的实现算法；
- 3. 在实验手册给出的两位数字乘法一个时钟移一位的基础上，实现满足一个时钟移动两位的乘法器，提升乘法效率，并在实验箱上验证。

三、 实验原理图

乘法器的原理是一系列的加法和移位操作，每一个时钟信号执行一次加法和移位操作，类似于 10 进制数的相乘，在计算机中，对于不同权值的，也是通过相同的操作运行的，即每左移 1 位相当于乘权值 2，右移一位相当于除权值 2。在两个 32 位定点数相乘时，我们首先对结果进行 64 位的拓展防止结果溢出，然后对于被乘数每个时钟信号左移 2 位（由于权值），乘数每个时钟信号右移两位（保证每次低两位存的都是我们本次需要计算的位），依次把每次计算结果相加保存，就完成了定点数的相乘运算。





(每时钟信号移动两位的迭代乘法原理图)

四、 实验步骤

Multiply 模块:

```
multiplicand <= {multiplicand[61:0], 2'b00}; //改成2位 在低位补两个0
```

1.第一处修改是，在每个时钟到来时，把被乘数算数左移 1 位改成算数左移 2 位，我们只需舍弃高两位，并在低位用两个 0 补位；

```
multiplier <= {2'b00, multiplier[31:2]};
```

2.第二处修改是，在每个时钟到来时，把乘数算数右移 1 位改成算数右移 2 位，我们只需舍弃低两位，并在高位用两个 0 补位；

// 部分积：乘数末位为1，由被乘数左移得到；乘数末位为0，部分积为0

```
wire [63:0] partial_product;
```

```
wire [63:0] partial_product1;
```

```
assign partial_product = multiplier[0] ? multiplicand : 64'd0;
```

```
assign partial_product1 = multiplier[1] ? multiplicand : 64'd0;
```

3.第三处修改是，由于现在每个时钟移动两位，即每次计算乘数的低两位与被乘数的乘法结果，然后传给加法器，那么我们需要两位来存储乘数的低两位，只需选择乘数的低

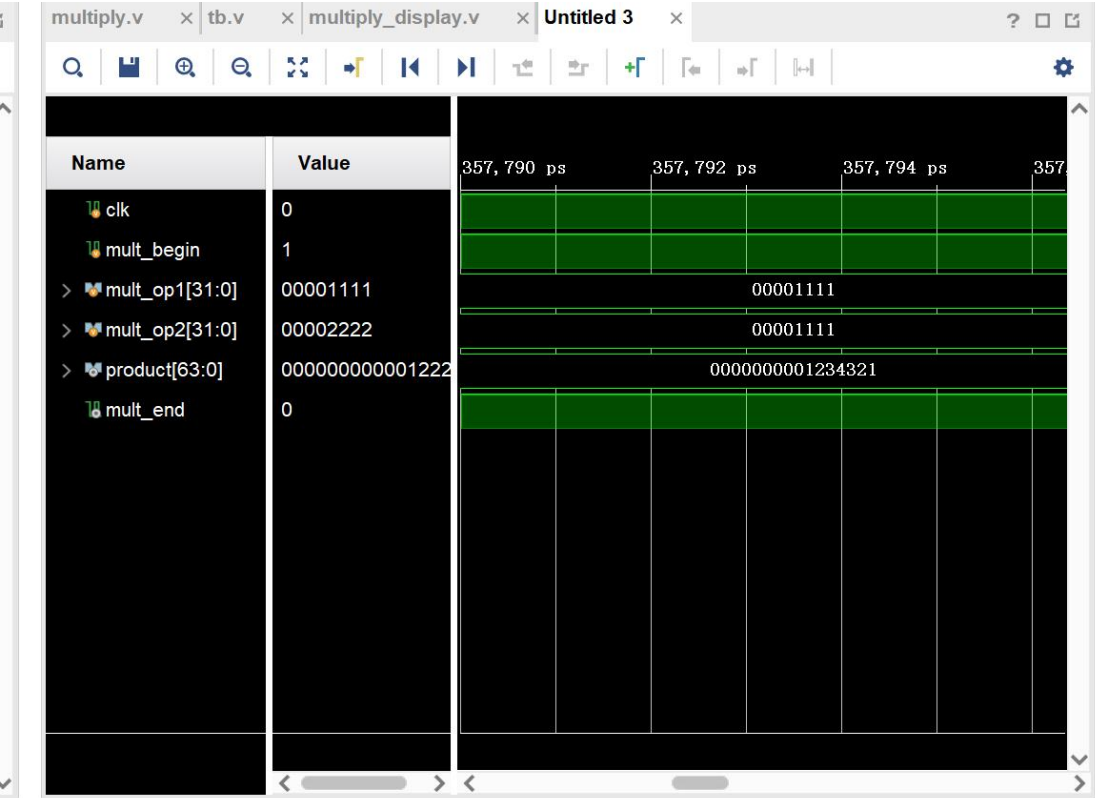
二位，multiplier[0]和 multiplier[1],partial_product 和 partial_product1 用来分别存储这两位对应的部分积,如果为 0 则对应存储 64 位 0，否则存储相应的被乘数运算结果；

```
g11
    product_temp <= product_temp + partial_product + 2 * partial_product1;
d
```

4.第四处修改是，在加法处作修改，每次运算结果在原来数值的基础上加上最低位部分积和次低位部分积的二倍（由于权重为 2 的 1 次方），即可得到新的结果。通过实现移位 1 位到移位 2 位，使得原来需要 32 个时钟信号完成的乘法任务，我们只需 16 个信号即可完成，显著提高了效率。

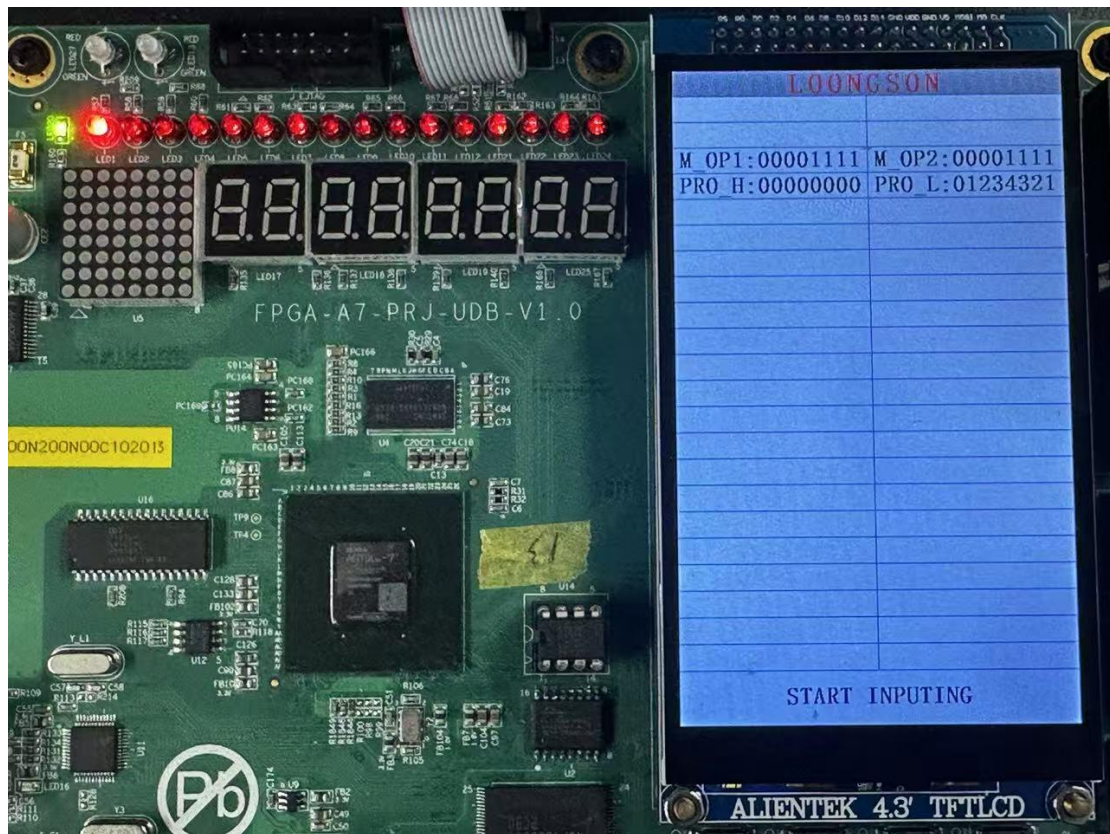
五、 实验结果分析

1.仿真结果



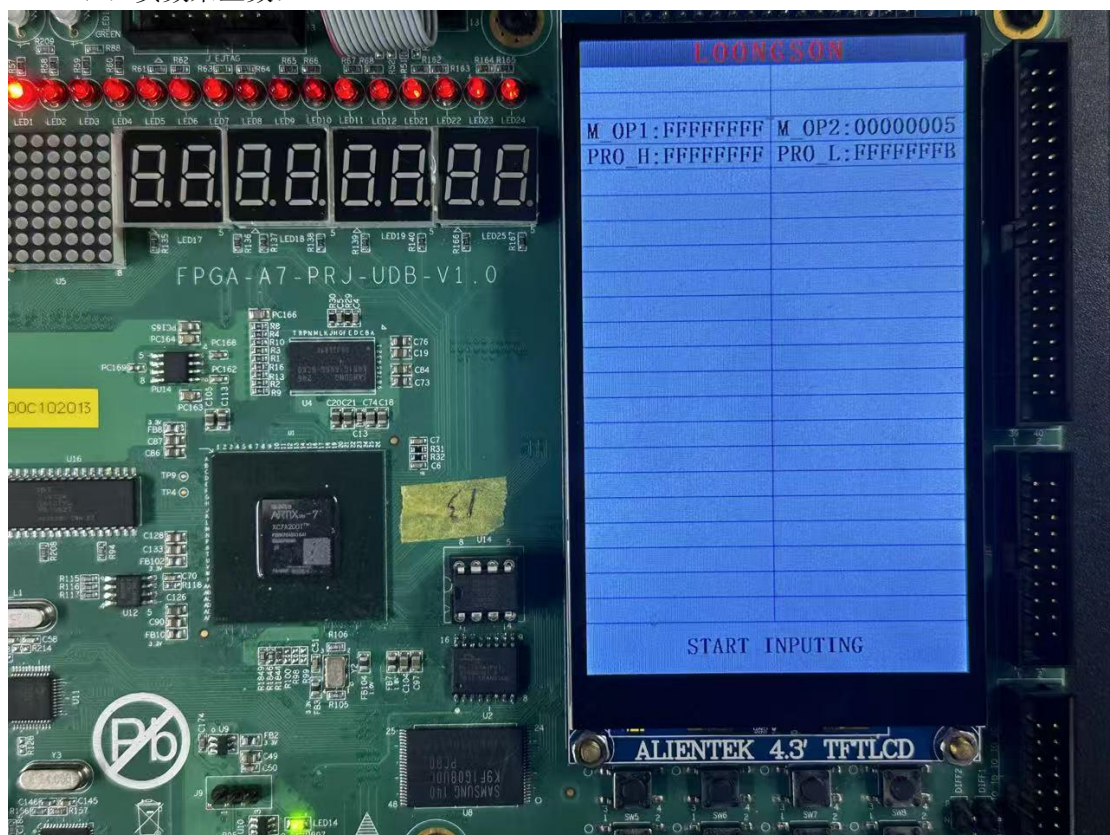
如图所示，两个 32 位数 00001111*00001111 得到 64 位的（高位 0）01234321，答案正确，说明乘法器的逻辑没有问题；

2. 实验箱运行结果



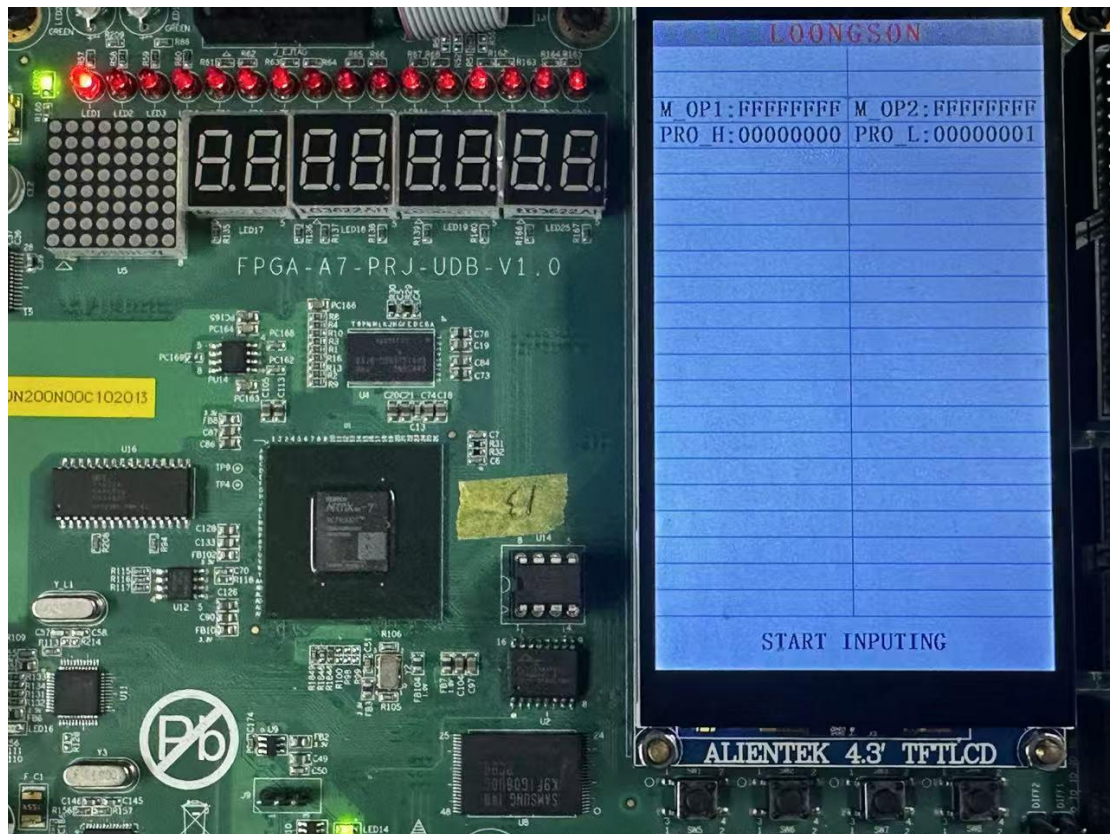
在实验箱上验证仿真模拟的数据，完全正确：

(1) 负数乘正数：



FFFFFFF 转成真值为-1，00000005 为 5，fffffffffffffb 表示-5，即 $-1 \times 5 = -5$ ，正确；

(2) 负数乘负数：



这里为 $-1 * -1 = 1$ ，正确，综上，我们的乘法器修改完毕，完全正确。

六、总结感想

- 1.能够熟练掌握 vivado 的使用以及如何去编写 Verilog 语句；
- 2.了解了如何设计一个乘法器，并且了解了迭代乘法的原理；
- 3.对移位操作有了更好的认识；
- 4.通过修改代码，实现了从一个时钟信号移动 1 位到一个时钟信号移动 2 位，优化了乘法器，显著提高了效率；
- 5.熟练了实验箱的使用方法。