**组成原理实验课程第 2 次实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 迭代乘法器 | | | 班级 | 李涛老师 |
| 学生姓名 | 申宗尚 | 学号 | 2213924 | 指导老师 | 董前琨 |
| 实验地点 | 实验A306 | | 实验时间 | 2024.3.28 | |

1. **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。

1. **实验内容**

1. 请结合实验指导手册中的实验二(定点乘法器实验)完成性能改进，不在是 原始的最长 32 个时钟周期完成乘法，注意以下几点:

2. 本次主要修改multipler,v模块，建议从两位乘开始进行，此外还有华莱士 树等高级优化方式，鼓励大家尝试。

3. 实验报告中需要补充原理图，并对原理图进行解释说明。原理图参照图3.2 进行修改，建议使用 visio 画图(别的画图软件也可，不会画图的可以手绘 然后照片放报告里面)。

4. 实验报告中需要有仿真结果(波形截图)，并针对图中的数据解释说明，还需 要有实验箱上箱验证的照片，同样，针对照片中的数据也需要解释说明。

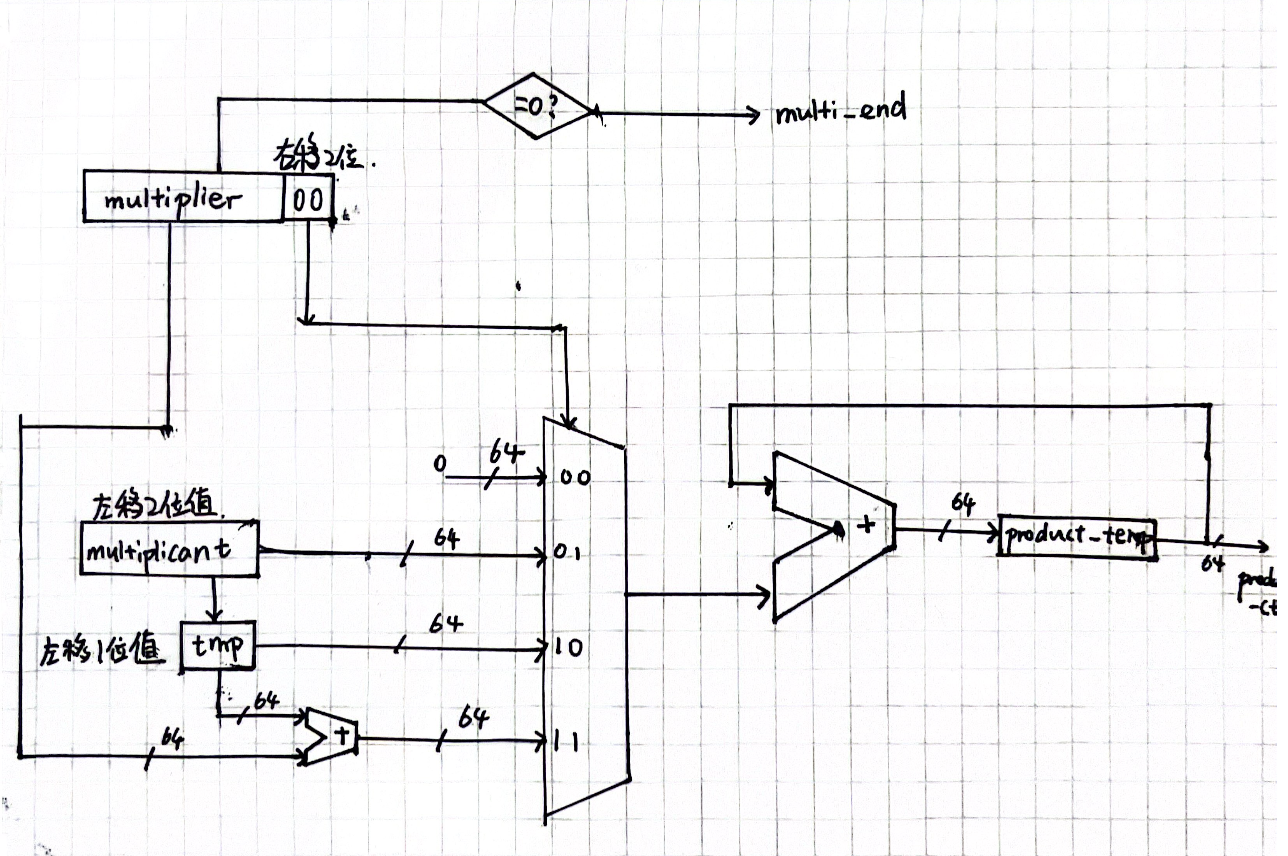
5. 实验报告模板参考百度云盘文件，注意提交截至时间为4月12日下午18: 00。

1. **实验原理图**

图示, 示意图

描述已自动生成

说明：顶层模块(multiply.v, lcd\_module.v, multiply\_display.v)等实现原理图与手册中相同，对其乘法器进行修改，修改后如下图所示：



1. **实验步骤**

该实验的代码工程如下图所示：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

由顶层模块multiply\_display实现，其中包含multiply模块（乘法器实现）。

由于本次实验是对迭代乘法器的改进，因此只涉及对乘法器模块的修改，顶层模块不需要进行修改。

在原本的乘法器实现中，每一个时钟周期乘数移动一位，如果完成 32 位二进制数的乘法需要 32 个时钟周期，改进后的乘法器每次移动两位，使得完成乘法的过程只需要 16 个时钟周期，实现性能的优化。时序逻辑部分代码对于移位操作的代码修改如下：

文本

描述已自动生成

对与部分积的赋值共分为四种情况，分别为 multiplier 最低两位为 00,01,10,11 的 时候：

* + - 1. 若为00，部分积为0
      2. 若为01，部分积为被乘数
      3. 若为10，部分积为被乘数的2倍（左移1位）
      4. 若为11，部分积为被乘数的3倍，（被乘数左移1位再加上被乘数）

代码实现如下:

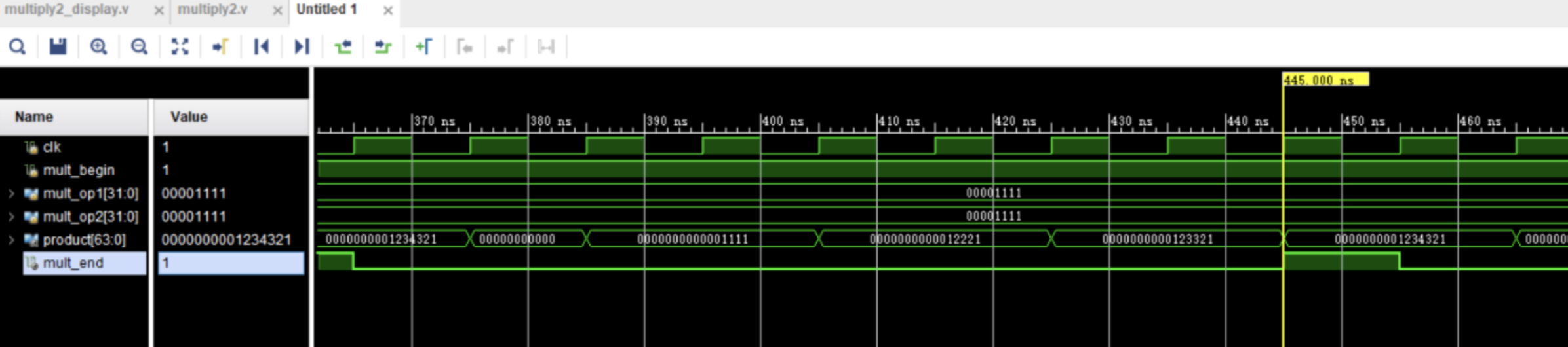
文本

描述已自动生成

同时，更改累加器，将partial\_product1与partial\_product2都累加到结果上。

1. **实验结果分析**

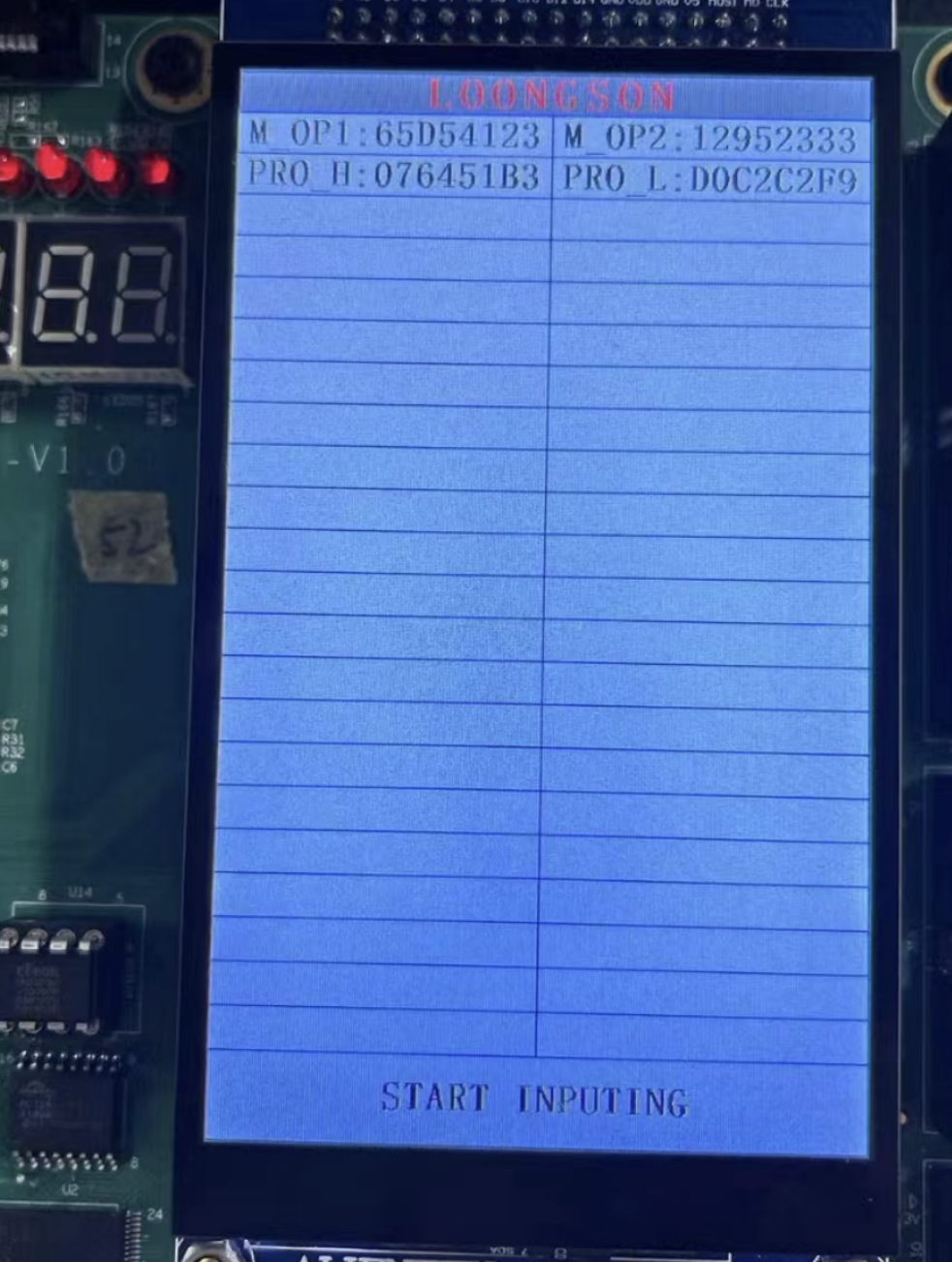
在vivado中运行仿真代码文件，得到以下波形结果：



00001111\*00001111=000000001234321

运算结果正确。

上箱验证：



在电脑的计算器上验证结果如下：

电子计算器

中度可信度描述已自动生成

从而，运算结果正确，实验成功。

1. **总结感想**
   * + 1. 进一步熟悉了verilog语言运用。
       2. 明白了乘法器的工作原理，并对其进行了改进，减短了时钟周期
       3. 深刻理解了迭代乘法器的工作原理，并对其进行优化，实现仿真、上机的结果成功s