**Booth算法乘法器 计算机组成与结构实验报告**

**姓名**：肖良寿  **学号**：1120200563

**班级**：07162001班 **手机**：15825065631

1. **实验题目**

设计参数化的32位booth算法乘法器模块，仿真验证正确性，并做改进分析。

1. **实验约束**

* 实验环境为Vivado2019.2
* 采用Verilog实现时使用结构化描述方式。

1. **算法原理**

Booth乘法算法是计算机中利用数的二进制补码形式来计算乘法的算法。可描述为：对于N位乘数Y，Booth乘法算法检查其二进制补码的最后一位和隐含的低位，命名为y-1，初始值为0。对于，考察yi和yi-1。当这两位数相同时，存放乘积的累加器P的值保持不变；当yi=0且yi-1=1时，被乘数乘以2i加到P中；当yi=1且yi-1=1时，从P中减去被乘数乘以2i的值。算法结束后，P中的数即为乘法的结果。

综上，Booth乘法的规则可总结如下：

1. 参与运算的数用补码表示；
2. 符号位参与运算；
3. 乘数最低位后面增加一位附加位yn+1，其初始值为0；
4. 乘数最低的两位作为判断位，其取值情况与对应的操作如下表所示：

|  |  |
| --- | --- |
| 判断位**YnYn+**1 | 操作 |
| 00 | 原部分积右移一位 |
| 01 | 原部分积加**[X]补**后右移一位 |
| 10 | 原部分积加**[-X]补**后右移一位 |
| 11 | 原部分积右移一位 |

1. 移位按补码右移的规则进行；
2. 共需n+1次累加，n次移位，第n+1次不移位。
3. **算法伪代码表示**

使用的寄存器：N, M, Q, Qres(Q右移后的残余位), PC(程序计数器)

**Step1**：加载寄存器的初始值

N=0(累加器)

**Step2**：检查的值value

IF value=00或11

转Step5；

IF value=01

转Step3；

IF value=10

转Step4；

**Step3**：执行，转Step5

**Step4**：执行

**Step5**：执行的算数位移和递减计数

**Step6**：检查计数器PC的值

IF PC=0

转Step7

ELSE 转Step2

**Step7**：停止计算，输出计算结果

1. **算法实现**

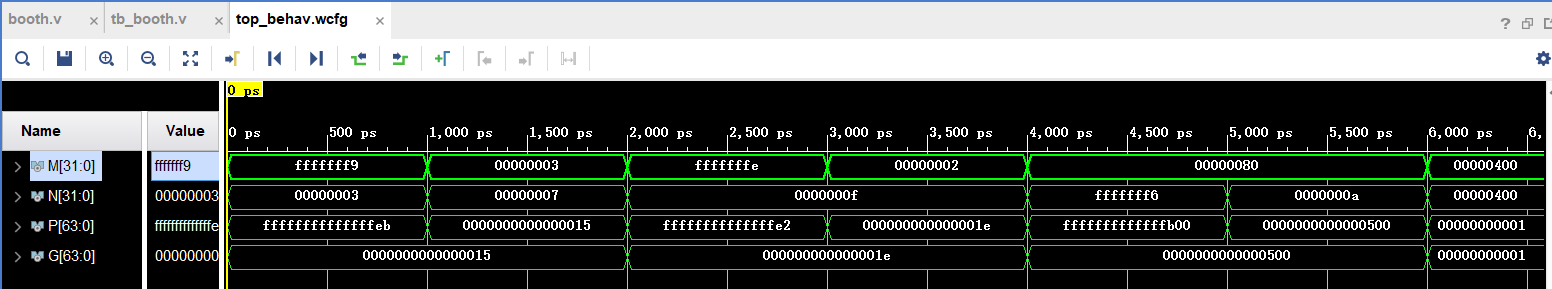
借助Verilog语言实现上一部分的算法伪代码。整体工程文件共编写两个模块：一个实现booth算法的主模块booth\_module，一个用于检测所编写的booth算法模块效果的测试模块test\_module。

1. **电路验证**
   1. **TestBench**

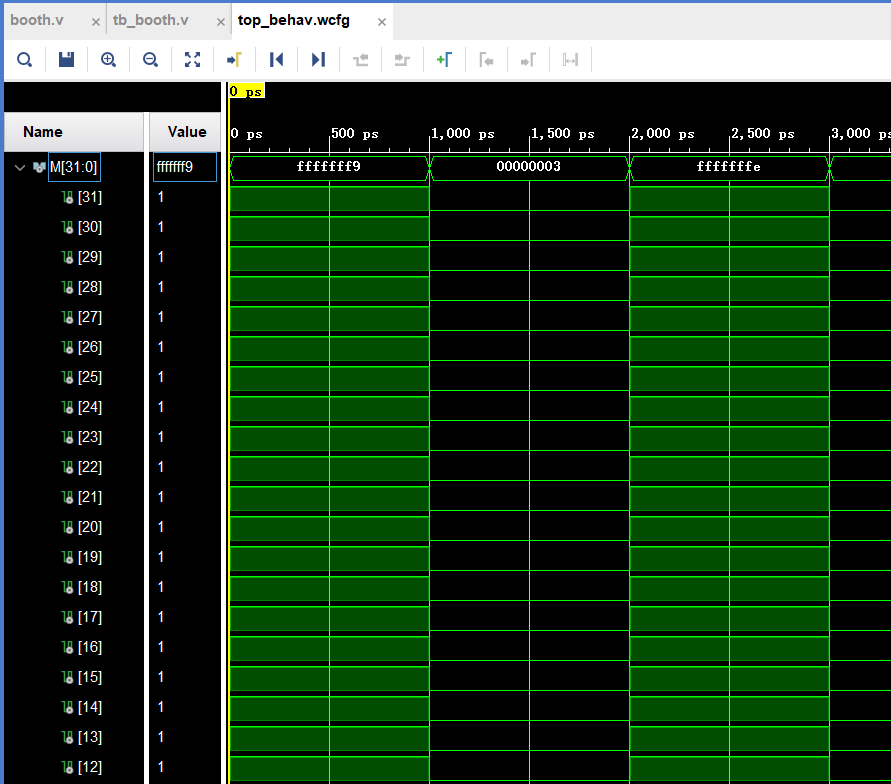
在Test Bench模块中例化上述的两个模块，分别处理输入输出信号。其中测试用的乘数与被乘数在test\_module中给出。

* 1. **仿真结果**

波形配置文件见于所提交项目的.wcfg文件，仿真波形如下：



图中M、N表示两个乘数，P为两数补码乘法得到结果的补码，G则表示乘数相乘结果的绝对值。实验中测试的部分几组数据为： -7\*3、7\*3、-2\*15、2\*15、-128\*10、128\*10。其中，M、N、P、G均为Array类型的数据，其每个位置的电位信号如下图：



1. **实验心得**

本次实验，帮助我加深了对Booth算法的流程的了解和掌握，通过波形图，直观展示了计算机中数值运算的过程。同时，实验中也遇到了一些困难，如Verilog语言与vivado软件的遗忘等。

总的看来，学期课程结束，收获颇丰，十分感谢老师的指导！