**深 圳 大 学 实 验 报 告**

**课 程 名 称： 计算机系统(3)**

**实验项目名称： MIPS64乘法器模拟实验**

**学 院： 计算机与软件学院**

**专 业： 计算机与软件学院所有专业**

**指 导 教 师： 刘刚**

**报告人： 何泽锋 学号： 2022150221 班级： 高性能特色班**

**实 验 时 间： 2024年10月10日~10月12日**

**实验报告提交时间： 2024年10月12日**

**教务处制**

**一、 实验目标：**

实际运用WinMIPS64进行试验，以期更了解WinMIPS64的操作；

更加深入地了解MIPS程序的语法；

深入地了解在计算机中乘法的实现以及加法与乘法之间的关系。

**二、实验内容**

按照下面的实验步骤及说明，完成相关操作记录实验过程的截图：

首先，我们使用加法操作设计一个不检测溢出的乘法操作；完成后，我们对此进行优化，以期获得一个可以对溢出进行检测的乘法操作。（100分）

**三、实验环境**

硬件：桌面PC

软件：Windows，WinMIPS64仿真器

**四、****实验步骤及说明**

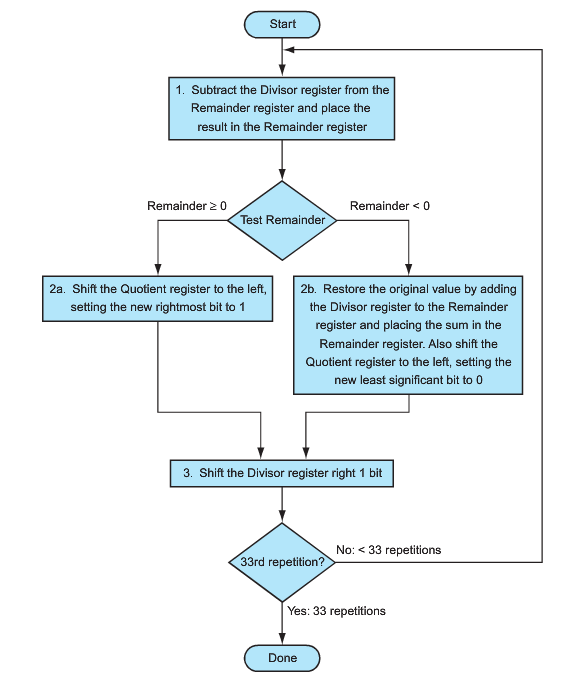
本次试验分为两个部分：第一部分、用加法器设计一个不考虑溢出的乘法器；第二部分、用加法器设计一个考虑溢出的乘法器（编程熟练的同学，也可以用除法器、浮点加法器等替代）。

1、忽略溢出的乘法器

首先，我们得了解乘法器如何由加法器设计得到，此处，我们以32位乘法为例。

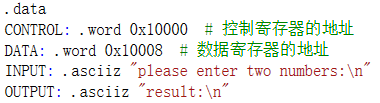
总共分为4步：

1. 测试乘数最低位是否为1，是则给乘积加上被乘数，将结果写入乘积寄存器；
2. 被乘数寄存器左移1位；
3. 乘数寄存器右移一位；
4. 判断是否循环了32次，如果是，则结束，否则返回步骤1。

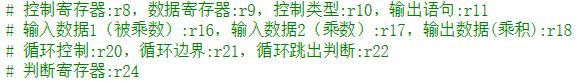


实现代码：

①首先需要先定义.data区，主要包括控制寄存器地址、数据寄存器地址以及终端运行时需要输出的提示语句



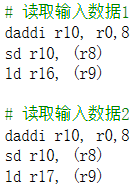
②接下来进行.text区的代码编写，此处将使用过的寄存器进行了注释，防止错误调用



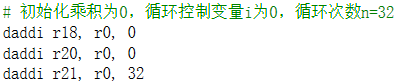
③首先是输出提示语句"please enter two numbers:"，需要设置控制寄存器的输出类型，此处为字符串需要设置为4，此外还需要将控制寄存器的地址、数据地址、字符串地址等传到寄存器中，具体代码如下所示：



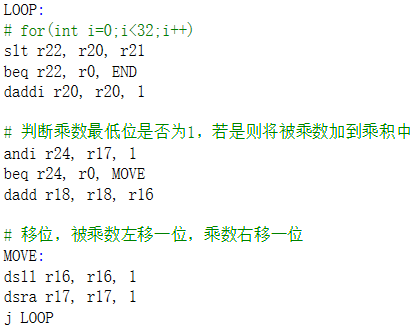
④读取终端中的输入数据，此处输入的数据是两个int类型数据，需要将控制器设置为8，然后从DATA取出读取的数据，两个数据需要执行两次，代码如下：



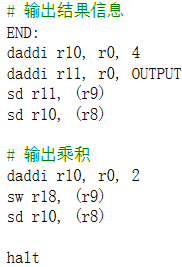
⑤初始化乘积、循环控制变量



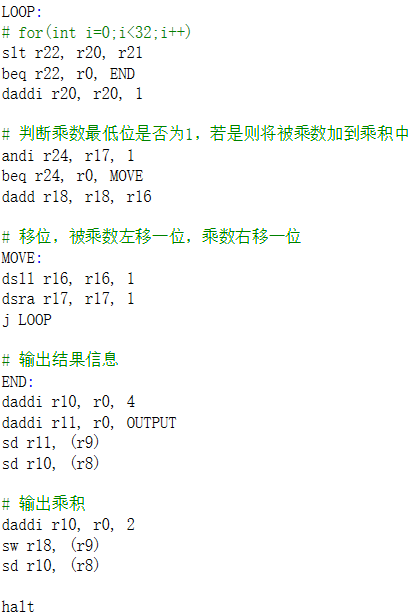
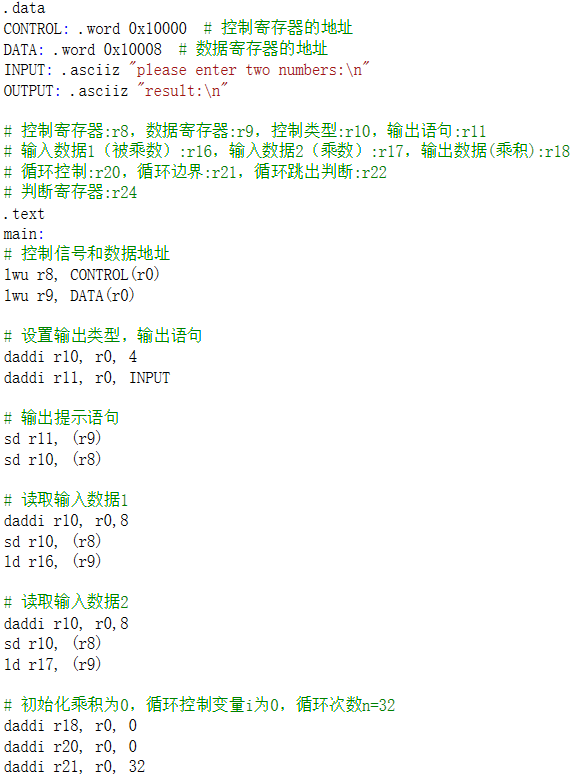
⑥使用循环实现乘法，每轮判断乘数的最低位是否为1，若是则将被乘数加到乘积中，然后将乘数右移1位，被乘数左移1位，需要注意移位操作每轮都需要进行，具体代码如下：



⑦输出结果信息，首先输出结果提示语句”result:”，然后输出运行的结果，代码如下：

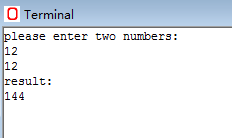


⑧完整代码展示：

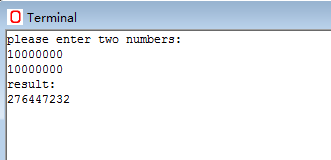


⑨运行显示运行结果的例子如下，由于我们这里展示的是忽略了溢出的乘法，所以结果有两种：1、小于32位；2、大于32位。

第一种情况截图：



第二种情况截图：



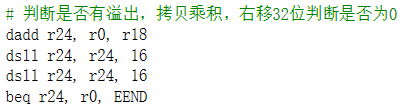
根据上面的程序代码和截图，我们可以很清楚的看出，当结果小于32位时，结果正常；当结果大于32位时，结果只截取了低32位的结果，而高32位的结果直接忽略掉了。

2、溢出提示的乘法器

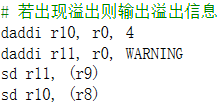
上述的程序，用加法实现了32位乘法，但是，其中，对溢出情况没有进行考虑是其中的弊端。这里，我们来完善上述的乘法器，使得该乘法器会在结果溢出时候提示。

实现代码：

①判断高32位是否为0，此处采用的方法为将乘积右移32位，然后将结果与0进行比较，若相等则说明没有溢出，否则溢出。需要注意指令集不支持直接右移32位，因此此处采用两次右移16位实现，代码如下：

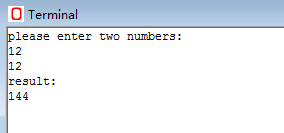


②若出现溢出则输出溢出信息，输出方式与前面输出提示语句一致，此处不过多说明

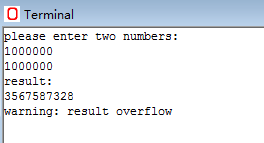


上述代码运行结果也有两个，一个是没有溢出的情况下的结果，一个是溢出了的情况下的结果。

首先，我们看没有溢出的情况结果：



结果正确，其次，我们看溢出的情况结果如何：



可以看到，当结果溢出时，程序会给出提示“warning：result overflow”。

4 结束语

本实验介绍了通过加法器来设计乘法器的原理，并且在编写该实验程序的时候，我们更加了解了：1、计算机乘法器工作原理的内容；2、进一步熟练MIPS的编程方法；3、WinMIPS64的使用方法。当然，如果想要更加深入的学习，我们也可以课外继续编写对除法的模拟。Perf软件的使用让学生初步熟悉性能测评的主要工具。

**五、实验结果**

实现了32位乘法器，同时也包含溢出判断，需要注意的是本次实验实现的乘法器只能用于判断正数之间的相乘，当输入负数时会出现错误，这是因为负数采用补码存储，高32位会有1，因此判断时会出现错误，误判为溢出。

**六、实验总结与体会**

通过本次实验对课程学习到的乘法器有了更深入的理解，也对移位的概念更加清晰，在代码实现上更加熟悉了如何使用Mips指令。在乘法的实现上了解了为什么乘法耗时远大于加法，因此对乘法的优化十分重要。本次实验主要实现了两种加法器，一种是最原始的乘法器，另一种是带有溢出判断的乘法器，实现上溢出判断也十分简单，只需通过判断移位后的值即可。

|  |
| --- |
| **指导教师批阅意见：**  **成绩评定：**  指导教师签字： 刘刚  2024年10月 日 |
| 备注： |