**深 圳 大 学 实 验 报 告**

**课程名称： 计算机系统(3)**

**实验项目名称： 实验二：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： 李琰**

**报告人： 叶茂林 学号： 2021155015 班级： 腾班**

**实验时间： 2023.10.18**

**实验报告提交时间： 2023.10.19**

**教务部制**

|  |
| --- |
| 实验目的与要求： 实验目的 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。 |
| 方法、步骤： 二、实验步骤及相关说明 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） |
| 实验过程及内容： 三、实验内容 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。  流程图如图1所示    图1  首先初始化一些数据，包括要输出的两个字符串提示信息和两个内容映射地址，如图2所示。    图2  然后写代码输出这两个字符串，其中输出pleas enter two numbers的代码如图3所示。    图3  然后需要获取被乘数和乘数，将CONTROL的值改为8，获取整数输入，如图4所示。    图4  然后开始乘法计算过程，如图5所示，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，否则跳到下一步，下一步是被乘数寄存器左移1位，乘数寄存器右移一位，最后判断是否循环了32次，如果是，则结束，否则返回第一步。    图5  最后把CONTROL改为2，输出乘积结果，如图6所示。    图6  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。  第一种情况截图如图7所示：    图7  第二种情况截图如图8所示：    图8  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。  其实，这个小优化是十分简单的，只需要对64位的寄存器中的高32位进行检测即可。当高32位为0时，说明结果没有溢出，否则，结果溢出。  首先在数据里加上一条提示字符串，如图9所示。    图9  将乘积寄存器进行算术右移32位取其高32，如图10所示，因不能一次移32位，所以分两次移位，然后判断是否为0，如果高32位为0说明没有溢出，如果不为0则说明已经溢出了，则输出提示字符串。    图10  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，如图11所示：    图11  结果正确，其次，我们看溢出的情况结果如何，如图12所示：    图12  可以看到，当结果溢出时，程序会给出提示“warning：result overflow”。 |

深圳大学学生实验报告用纸

|  |
| --- |
| 实验结论：  在本次实验中，我们首先使用MIPS汇编语言设计了一种不检测溢出的乘法操作。我们采用了一种比较简单的算法，每次判断乘数最低位是否为1，并对被乘数和乘积进行相应的处理。通过循环移位和加法操作，我们最终得到了正确的乘积结果。  然而，在这种操作下，我们并没有进行溢出检测。接着，我们对这个乘法操作进行了优化，以期获得一个可以对溢出进行检测的乘法操作。具体方法是对64位寄存器中的高32位进行检测，如果高32位为0，则说明结果没有溢出，否则，结果溢出。通过引入这个额外的判断步骤，我们成功地优化了乘法操作，使其能够检测并处理溢出情况。  通过本次实验，我深刻认识到了溢出检测的重要性。在实际程序中，溢出常常会导致程序错误和数据损坏，因此及时检测并处理溢出情况至关重要。此外，对MIPS汇编语言的深入学习和理解也对于优化程序和提高代码效率有着重要意义。  本次也让我更加熟悉了MIPS汇编语言的使用和特性。通过手动编写处理器指令，我更加深入地了解了计算机底层运行原理，并熟练掌握了MIPS汇编语言的基本语法和指令集。 |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。