**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： 罗秋明**

**报告人： 马庭俊 学号： 2018152102 班级： 数计班**

**实 验 时 间： 2020年11月11日星期三**

**实验报告提交时间： 2020年11月11日星期三**

**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**

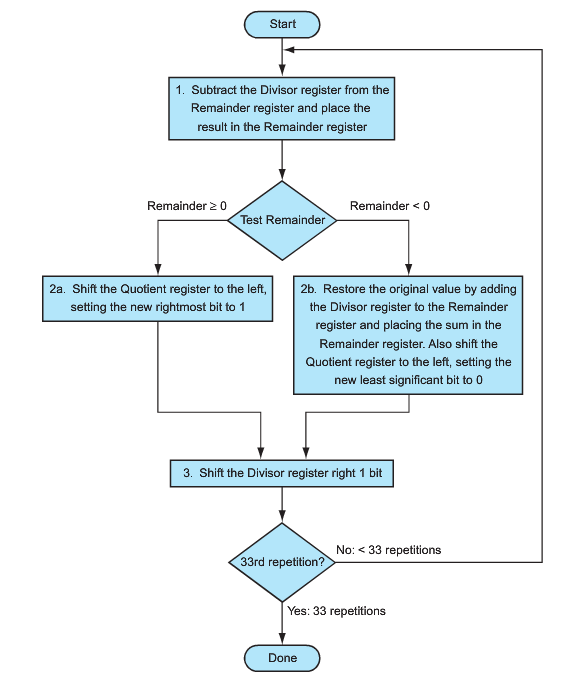
本次试验分为两个部分：第一部分、用加法器设计一个不考虑溢出的乘法器；第二部分、用加法器设计一个考虑溢出的乘法器（编程熟练的同学，也可以用除法器、浮点加法器等替代）。

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。

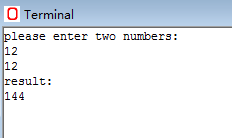
总共分为4步：

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。

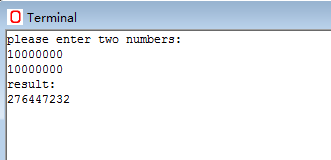


运行显示运行结果的例子如下，由于我们这里展示的是忽略了溢出的乘法，所以结果有两种：1、小于32位；2、大于32位。

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。

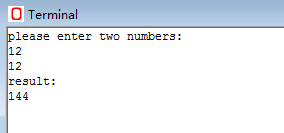
2、溢出提示的乘法器

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。

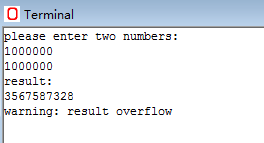
其实，这个小优化是十分简单的，只需要对64位的寄存器中的高32位进行检测即可。当高32位为0时，说明结果没有溢出，否则，结果溢出。

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**

**.data**

**Prompt: .asciiz "Please enter two numbers:\n"**

**ResultIs: .asciiz "the Result:\n"**

**Warning: .asciiz "Warning: result overflow!\n"**

**CR: .word32 0x10000**

**DR: .word32 0x10008**

**# space note!**

**.text**

**main:**

**daddi $a0, $R0, Prompt # output Prompt**

**jal printf\_str**

**lwu r21,CR(r0)**

**lwu r22,DR(r0)**

**daddi r24,r0,8 ;read input**

**sd r24,(r21)**

**lwu r16,(r22) ;multiplicand in R16**

**sd r24,(r21)**

**lwu r17,(r22) ;multiplier in R17**

**dadd $a0, r16, r0**

**dadd $a1, r17, r0**

**jal prod**

**daddi $a0, $R0, ResultIs**

**jal printf\_str**

**dadd $a0, $R0, $v0**

**jal printf\_integer**

**dsrl $v0, $v0, 16 # right shift 32 bits**

**dsrl $v0, $v0, 16**

**beqz $v0, end # if high == 0 --> end, else warning**

**daddi $a0, r0, Warning**

**jal printf\_str**

**end:**

**halt**

**prod: # return c = a \* b**

**dadd $v0, $zero, $zero # v0-->c a0-->a a1-->b**

**daddi $t0, $zero, 32**

**loopprod:**

**andi $t1, $a1, 1**

**beqz $t1, next # == 1 --> c = c + a**

**dadd $v0, $v0, $a0**

**next:**

**dsll $a0, $a0, 1**

**dsrl $a1, $a1, 1**

**daddi $t0, $t0, -1**

**bnez $t0, loopprod**

**jr $ra**

**printf\_str: # $a0 <--> str to be output**

**ld $t0, CR($R0) # t0 <--> CR**

**ld $t1, DR($R0) # t1 <--> DR**

**daddi $t2, $R0, 4**

**sd $a0, 0($t1) # str output**

**sd $t2, 0($t0) # to screen**

**jr $ra**

**printf\_integer: # $a0 <--> integer to be output**

**ld $t0, CR($R0) # t0 <--> CR**

**ld $t1, DR($R0) # t1 <--> DR**

**daddi $t2, $R0, 2**

**sd $a0, 0($t1) # signed number output**

**sd $t2, 0($t0) # to screen**

**jr $ra**

**五、实验总结与体会**

**遇到的问题，mips当中有个非常奇怪的错误，以下是描述：**

**.data # right code**

**str1: .asciiz "please enter two numbers:\n" # when I change 'please' to 'plz', something wrong happends! why?**

**str2: .asciiz "the Result:\n"**

**str3: .asciiz "Waring:result overflow!\n"**

**CONTROL: .word32 0x10000**

**DATA: .word32 0x10008**

**.text**

**main:**

**lwu r21,CONTROL(r0)**

**lwu r22,DATA(r0)**

**daddi r24,r0,4 ;set string output**

**daddi r1,r0,str2 ;get address of str1**

**sd r1,(r22) ;wirte DATA**

**sd r24,(r21) ;print**

**halt**

**当运行以上代码时，会正常输出字符串：**



图 1，正常输出字符串

**如图 1所示，该代码会正常输出字符串”the Result”，但是当我做如下修改：**

**.data # right code**

**str1: .asciiz "plz enter two numbers:\n" # when I change 'please' to 'plz', something wrong happends! why?此处为修改部分，仅仅将please替换成plz**

**str2: .asciiz "the Result:\n"**

**str3: .asciiz "Waring:result overflow!\n"**

**CONTROL: .word32 0x10000**

**DATA: .word32 0x10008**

**.text**

**main:**

**lwu r21,CONTROL(r0)**

**lwu r22,DATA(r0)**

**daddi r24,r0,4 ;set string output**

**daddi r1,r0,str2 ;get address of str1**

**sd r1,(r22) ;wirte DATA**

**sd r24,(r21) ;print**

**halt**

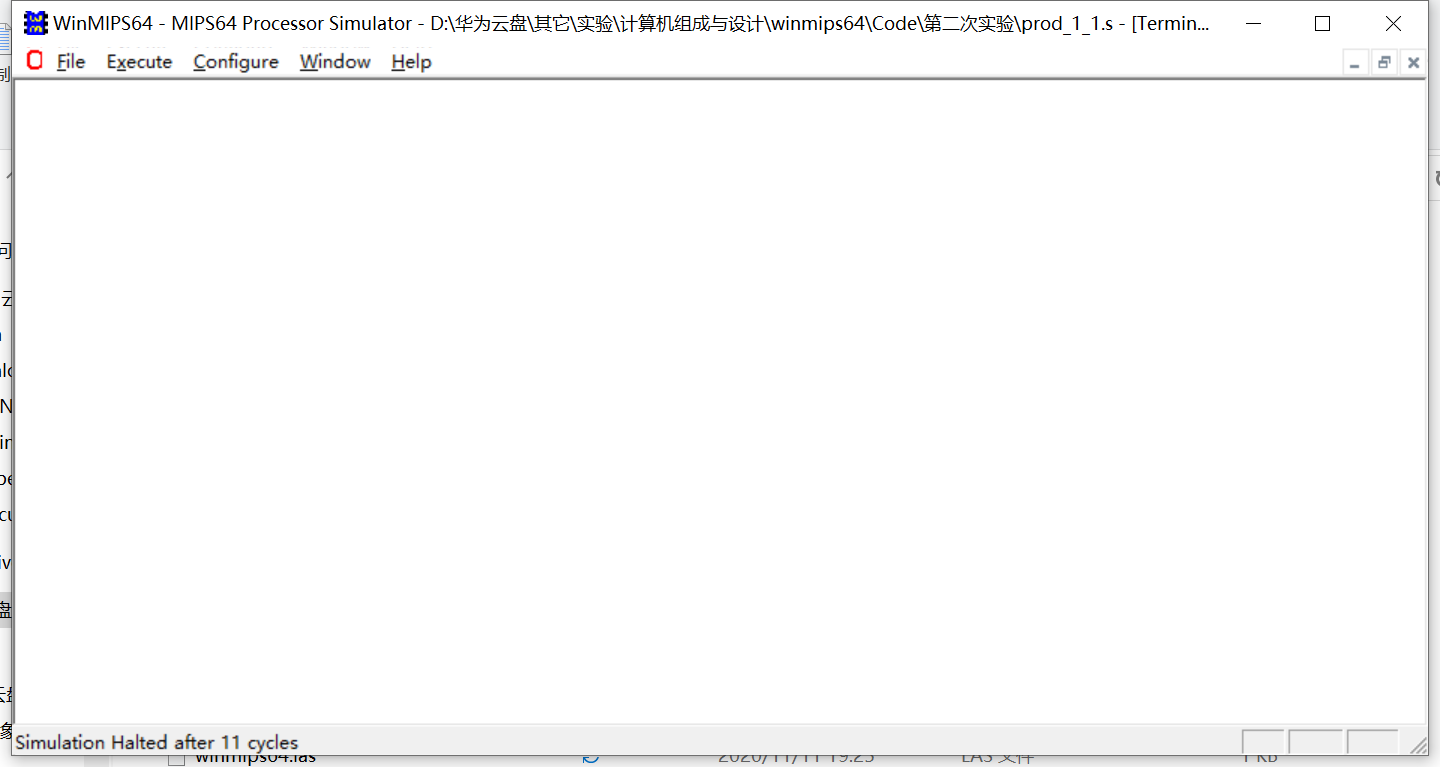


图 2，仅仅修改了字符串里面的内容，输出就异常了

**就会出现图 2的奇怪现象，这花费了我大量时间寻找错误，在**[**stackoverflow**](https://stackoverflow.com/questions/64736554/can-not-output-a-string-in-winmips64-normally)**上询问，有一个可能的错误原因，没有对齐，lwu指令必须使用以某个特定数为倍数的地址，可以使用.align简单对齐一下。不过，我尝试了.align，没有什么效果。**

**综上，我想询问一下这个这个问题。**

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