

重庆大学课程设计报告

课程设计题目: MIPS 浮点运算的整数实现

学 院: 计算机学院

专 业 班 级: 计算机科学与技术 01 班

年 级: 2018

学 生: 王智超

学 号: 20181819

完 成 时 间: 2020 年 6 月 25 日

成 绩: 90

指 导 教 师: 钟将

重庆大学教务处制

项目	分值	优秀 100 > x ≥ 90	良好 90 > x ≥ 70	中等 80 > x ≥ 70	及格 70 > x ≥ 60	不及格 x < 60	评分
		参考标准					
学 习 态度	15	学习态度认真,科学作风严谨,严格保证设计时间并按任务书中规定的进度开展各项工作	学习态度比较认真,科学作风良好,能按期圆满完成任务书规定的任务	学习态度尚好,遵守组织纪律,基本保证设计时间,按期完成各项工作	学习态度尚可,能遵守组织纪律,能按期完成任务	学习马虎,纪律涣散,工作作风不严谨,不能保证设计时间和进度	
技 术 水 平 与 实 际 能 力	25	设计合理、理论分析与计算正确,实验数据准确,有很强的实际动手能力、经济分析能力和计算机应用能力,文献查阅能力强、引用合理、调查调研非常合理、可信	设计合理、理论分析与计算正确,实验数据比较准确,有较强的实际动手能力、经济分析能力和计算机应用能力,文献引用、调查调研比较合理、可信	设计合理,理论分析与计算基本正确,实验数据比较准确,有一定的实际动手能力,主要文献引用、调查调研比较可信	设计基本合理,理论分析与计算无大错,实验数据无大错	设计不合理,理论分析与计算有原则错误,实验数据不可靠,实际动手能力差,文献引用、调查调研有较大的问题	
创新	10	有重大改进或独特见解,有一定实用价值	有较大改进或新颖的见解,实用性尚可	有一定改进或新的见解	有一定见解	观念陈旧	
论 文 (计 算 书、图 纸) 撰 写 质 量	50	结构严谨,逻辑性强,层次清晰,语言准确,文字流畅,完全符合规范化要求,书写工整或用计算机打印成文;图纸非常工整、清晰	结构合理,符合逻辑,文章层次分明,语言准确,文字流畅,符合规范化要求,书写工整或用计算机打印成文;图纸工整、清晰	结构合理,层次较为分明,文理通顺,基本达到规范化要求,书写比较工整;图纸比较工整、清晰	结构基本合理,逻辑基本清楚,文字尚通顺,勉强达到规范化要求;图纸比较工整	内容空泛,结构混乱,文字表达不清,错别字较多,达不到规范化要求;图纸不工整或不清晰	

指导教师评定成绩:

指导教师签名:

MIPS 浮点运算的整数实现

王智超

1 小组分工说明

王智超:负责实现方案。

负责使用 Mars 编写汇编,用整数完成浮点数的加法。

负责课程设计报告的填写。

2 设计方案

(1) 总体设计思路

首先,我们要用到 $\$s0, \$s1, \$s2, \$s3, \$s4, \$s5$,因此需要将其保存到堆栈中;接着将存储在 $\$a0$ 的第一个浮点数按照 IEEE754 分解成三部分符号位、指数、尾数,通过左右移位操作将三部分提取出来,并分别存储在 $\$s0, \$s1, \$s2$,第二个浮点数 $\$a1$ 同样操作,存储在 $\$s3, \$s4, \$s5$,同时分别在尾数的最高位添加上隐藏的 1,方便进行运算。其次,进行两个浮点数的加法运算。

第一步,要进行指数对齐,比较两个数的指数,将指数较小的数进行右移,直到其指数与较大的相匹配。

第二步,通过符号位,判断两个数是否为负数,若是负数,将其转化成补码形式,方便进行带负数的加法运算(也即减法),然后将尾数相加得到一个结果,同样若结果为负数,要取反加 1,然后将 $\$a2$ (存储结果的寄存器)的最高位(符号位)置为 1,指数位置为跟指数较大的一样。

第三步,要对结果进行规格化,要求表示为小数点左边只有一位非零数 1 的形式,这就要找到尾数相加的结果中第一个 1 出现的位置,然后将结果进行左移或者右移操作,右移时增大指数,左移时减小指数。

第四步,对尾数进行舍入,去掉最高位的隐藏位 1 后,将结果截断成 23 位,赋给 $\$a2$ 的低 23 位,同时,给 $\$a2$ 的指数位也进行赋值,得到浮点数相加的结果。

最后从堆栈中恢复使用过的寄存器。

对于浮点数的整数运算检测,我们随机产生两个浮点数,同时用 mips 的 `add.s` 指令进行计算,将两者的结果进行比较,如果一致,则认为计算正确。

(2) yourfunc 模块设计

```

59 #进行指数对齐
60 slt $t0, $s1, $s4
61 beq $t0, $0, shift_right_second #若$t0等于0, 则说明第二个浮点数的指数较小, 需要进行右移对齐
62 sub $t1, $s4, $s1
63 srlv $s2, $s2, $t1 #第二个浮点数的尾数右移 $t1位
64 move $s1, $s4 #指数赋值
65 j change_to_complement
66
67 shift_right_second:
68 sub $t1, $s1, $s4
69 srlv $s5, $s5, $t1 #第二个浮点数的尾数右移 $t1位
70 move $s4, $s1 #指数赋值
71
72 change_to_complement:
73 #将两个数转换成补码形式, 便于运算
74 beq $s0, $0, label1
75 nor $s2, $s2, $0 #取反
76 addi $s2, $s2, 1 #加1
77
78 label1:
79 beq $s3, $0, label2
80 nor $s5, $s5, $0 #取反
81 addi $s5, $s5, 1 #加1
82
83 label2:
84 add $t0, $s2, $s5 # $t0 存储着结果的尾数
85 move $a2, $0 #给最后结果赋初始值0
86

```

图 1: 指数对齐截图

```

93 label3:
94 #循环查找第一个为1的位置
95 addi $t3, $0, 32
96 find_first_position:
97 beq $t3, $0, out
98 addi $t3, $t3, -1
99 srlv $t4, $t0, $t3
100 bne $t4, 1, find_first_position
101
102 #结果进行规格化
103 out:
104 sub $t3, $t3, 23
105 slt $t4, $0, $t3
106 beq $t4, $0, shift_left_fraction #若$t4=0, 说明$t3是负数, 即将尾数左移, 指数加上移动的位数
107 srlv $t0, $t0, $t3 #尾数右移
108 add $t5, $s1, $t3 #指数加上正的移动位数
109 j result
110
111 shift_left_fraction:
112 sub $t3, $0, $t3 #将$t3转化为正数
113 sllv $t0, $t0, $t3 #尾数左移
114 sub $t5, $s1, $t3 #指数减去移动位数
115
116 result:
117 andi $t0, $t0, 0x7fffff #将隐藏位去掉
118 add $a2, $a2, $t0 #尾数域赋值
119 sll $t2, $t5, 23
120 or $a2, $a2, $t2 #指数域赋值

```

图 2: 结果规格化截图

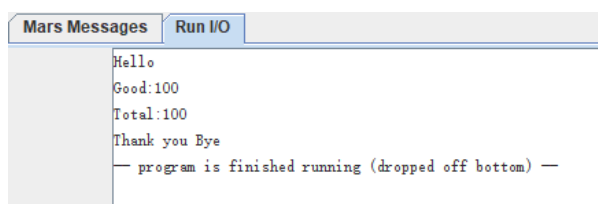
浮点数的加法重点部分主要是指数的对齐和结果的规格化,指数对齐要进行比较,将指数较小的数进行右移,直到其指数与较大的相匹配。而对于结果的规格化,要首先将计算的结果从补码形式转化为原码,然后通过移位,使得二进制小数点左边只有一位非零数,右移时增大指数,左移时减小指数。

3 设计结果

(1) 总体设计思路

```
133 getrandom:
134     li $v0, 43          #getrandom
135     addi $a0, $0, 10    #
136     syscall
137     sub $sp, $sp, 4
138     s.s $f0, ($sp)
139     lw $a0, ($sp)
140     addi $a0, $a0, 0x2000000
141     andi $a0, $a0, 0xffff000
142     sw $a0, ($sp)
143     l.s $f0, ($sp)
144     addi $sp, $sp, 4
145     jr $ra
```

图 3: 随机产生浮点数输入截图



The screenshot shows a window titled "Mars Messages" with a "Run I/O" button. The output text is as follows:

```
Hello
Good:100
Total:100
Thank you Bye
— program is finished running (dropped off bottom) —
```

图 4: 输出结果截图

结果分析:通过随机产生两个浮点数输入,用 mips 的指令 add.s 进行计算,同时用我们的 yourfunc 模块进行浮点数的整数运算,得到两个结果,将两个结果进行比较,如果相同,则认为该次计算正确,good 的数量加 1。总共(total)产生 100 次输入,结果显示 good 的数量为 100,在一定精度范围内,通过了测试程序,说明 yourfunc 模块正确,可以完成浮点数运算的整数实现。

4 总结(可选)

- (1) 通过这次课程设计,对 mips 的汇编指令的使用更加熟练,并且熟练掌握了 IEEE754 中单精度浮点数的表示和计算规则。同样,对用汇编语言编写代码的繁杂性和 debug 的复杂性有了更深的认识,

- (2) 浮点数的整数实现在此次设计中要求的精度不是很高,因此没有考虑保护位、舍入位、粘贴位等增加尾数舍入精度的位,但正确通过了测试程序。

5 参考文献

- [1] Patterson D , Hennessy J , 帕特森, et al. 计算机组成与设计: 硬件/软件接口 [M]. 机械工业出版社, 2007.
- [2] Larus J R . Assemblers, Linkers, and the SPIM Simulator[J]. Computer Organization and Design, 1994, 13(2):A-2–A-72.