**《计算机组成原理》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **年级、专业、班级** | | **2021级 计算机科学与技术（卓越）2班** | | | **姓名** |  |
| **实验题目** | **Mips汇编程序设计** | | | | | |
| **实验时间** | **2022.4.18** | | **实验地点** |  | | |
| **实验成绩** |  | | **实验性质** | **□验证性 🗹设计性 □综合性** | | |
| 教师评价：  □算法/实验过程正确； □源程序/实验内容提交 □程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； □报告规范；  其他：  评价教师签名： | | | | | | |
| 1. 实验目的   1. 项目目标  (1) 深入掌握二进制数的表示方法以及不同进制数的转换；  (2) 掌握二进制不同编码的表示方法,掌握IEEE 754 中单精度浮点数的表示和计算;  (3) 能够应用MIPS汇编进行编程。 | | | | | | |
| 二、实验项目内容  假设没有浮点表示和计算的硬件，用软件方法采用仿真方式实现IEEE 754单精度浮点数的表示及运算功能，具体要求如下：  (1) 程序需要提供人机交互方式（字符界面）供用户选择相应的功能；  (2) 可接受十进制实数形式的输入，在内存中以IEEE 754单精度方式表示，支持以二进制和十六进制的方式显示输出；  (3) 可实现浮点数的加减(或者乘除)运算；  (4) 使用MIPS汇编指令，但是不能直接使用浮点指令，只能利用整数运算指令来编写软件完成。  （5） 设计报告中给出程序的需求分析和关键算法的流程图，提交的代码注释比例>40%,注释语义清晰。 | | | | | | |
| 1. 实验设计思路   **需求分析：**  本程序首先需要用户使用键盘输入两个操作数，并指定运算符和输出格式。程序根据用户输入的两个操作数和运算符得到结果，并根据用户指定的输出格式将结果输出在屏幕上。  **算法流程图：**  取出用IEE754存储的浮点数的尾数部分、指数部分和符号位  比较两个数的绝对值大小，同时判断减法结果的符号  给尾数加上前导1，并通过右移操作将绝对值较小数的尾数向绝对值较大数对齐  判断尾数之和是否达到2或者尾数之差是否小于1  通过移位操作对尾数运算结果进行规格化处理  用整数运算指令求尾数的和（差）  去掉尾数的前导1，并将其和较大的指数、运算结果的符号位拼接起来  将尾数部分转换成相应的字符串并在前面加上“1.”  分别取出浮点数的三个部分  判断输出格式  输出尾数部分的字符串+2^指数部分对应的十进制数  如果选择十进制输出，则直接调用syscall指令  结束 | | | | | | |
| 四、实验过程或算法（源程序）  .data  str1: .asciiz "Please input in format: num1+Enter, num2+Enter, ope-syms: \n"  str2: .asciiz "\nThe answer is "  str3: .asciiz "\ninput 'B, D' to select output format: \n"  str4: .asciiz "\nError: You can only input B or D to select output format!"  str5: .asciiz " x 2^"  syms1: .asciiz "+"  syms2: .asciiz "-"  ans: .space 64  input1: .space 32  input2: .space 32  .text  .globl main  main:  li $v0, 4 # 将系统调用4存储在v0寄存器中  la $a0, str1 # 将字符串的地址存储在a0寄存器中  syscall # 调用系统调用4，输出字符串  li $v0, 6 #输入浮点数  syscall  mov.s $f1, $f0 # 将返回值从 $f0 中传递给目标浮点寄存器 $f2  syscall  mov.s $f2, $f0 # 将返回值从 $f0 中传递给目标浮点寄存器 $f2  li $v0 12 #输入运算符  syscall  move $t1 $v0    mfc1 $t5, $f1 #将操作数从浮点数寄存器传到整数寄存器  mfc1 $t6, $f2  bne $t1 45 else #判断运算符号  xori $t6 $t6 0x80000000  else:  andi $s1, $t5, 0x007fffff #取尾数  andi $s2, $t6, 0x007fffff  ori $s1, $s1, 0x00800000 #在尾数的第24位加上前导1  ori $s2, $s2, 0x00800000  andi $s3, $t5, 0x7f800000 #取指数域  andi $s4, $t6, 0x7f800000  andi $s5, $t5, 0x80000000 #取最高位，即符号位  andi $s6, $t6, 0x80000000  andi $t5 $t5 0x7fffffff  andi $t6 $t6 0x7fffffff  slt $t0, $t5, $t6 #判断两个指数的大小  bne $t0, $zero, Label2  #(s3>)=(s4)  Label1: andi $t7, $s5, 0x80000000  add $t7, $t7, $s3 #将指数和符号位存储在t7  sub $s0, $s3, $s4  ror $s0, $s0, 23  sub $t3, $s1, $s2  #循环右移， 调整指数较小的数的尾数  Loop1:  beq $s0, $zero, temp1  srl $s2, $s2, 1  addi $s0, $s0, -1  j Loop1  temp1: sub $t3, $s1, $s2  j exit  #(s3)<(s4)  Label2: andi $t7, $s6, 0x80000000  add $t7, $t7, $s4 #将指数和符号位存储在t7  sub $s0, $s4, $s3  ror $s0, $s0, 23  sub $t3, $s2, $s1  #循环右移,调整指数较小的数的尾数  Loop2: beq $s0, $zero, temp2  srl $s1, $s1, 1  addi $s0, $s0, -1  j Loop2  temp2: sub $t3, $s2, $s1  j exit  exit:  xor $t1,$s5, $s6  bne $t1, $zero, Label4 #判断是否异号  #同号  Label3:  add $s1, $s1, $s2  andi $t2, $s1, 0x01000000  beq $t2, $zero, else1  srl $s1, $s1, 1  addi $t7, $t7, 0x00800000  else1: andi $s1, $s1, 0xff7fffff #舍去前导1  or $t5, $s1, $t7 #将尾数和指数拼接  mtc1 $t5, $f12 #将结果传回浮点数寄存器  j Exit  #异号  Label4: bne $t3 $zero else2 #判断结果是否为0  j Exit    else2: andi $t2, $t3, 0x00800000  bne $t2, $zero, else3  loop3: bge $t3, 0x00800000, else3  sll $t3, $t3, 1  subi $t7, $t7, 0x00800000  j loop3    else3: andi $t3, $t3, 0xff7fffff #舍去前导1  or $t5,$t3, $t7 #将尾数和指数拼接  mtc1 $t5, $f12  j Exit  Exit: li $v0 4  la $a0 str3  syscall  li $v0 12  syscall  move $t1 $v0    beq $t1 66 binary  beq $t1 68 decimal  li $v0 4  la $a0 str4  syscall  j Exit    binary:  la $s0 ans  mfc1 $t5 $f12  andi $t6 $t5 0x7f800000 #取指数域  andi $t7 $t5 0x80000000 #取最高位符号位  ror $t6 $t6 23  subi $t6 $t6 127    li $t8 49 #在ans字符串中加上前导1  sb $t8 ($s0)  addi $s0 $s0 1  li $t8 46 #在ans字符串中加'.'  sb $t8 ($s0)  addi $s0 $s0 1    li $t2 0  loop4: andi $t8 $t5 0x00400000  srl $t8 $t8 22  sll $t5 $t5 1  addi $t8 $t8 48  sb $t8 ($s0)  addi $s0 $s0 1  addi $t2 $t2 1  bne $t2 23 loop4    li $v0 4  la $a0 str2  syscall  beq $t7 0x80000000 else4 #判断结果的正负号  la $a0 syms1  j exit2  else4: la $a0 syms2  exit2: syscall  la $a0 ans #输出尾数部分  syscall  la $a0 str5  syscall  li $v0 1 #输出指数部分  move $a0 $t6  syscall  li $v0, 10 # 调用exit系统调用  syscall # 执行系统调用    decimal:  li $v0 4  la $a0 str2  syscall #输出结果  li $v0, 2  syscall  li $v0, 10 # 调用exit系统调用  syscall # 执行系统调用 | | | | | | |
| 五、实验结果及分析和（或）源程序调试过程  以十进制形式输出结果    以二进制形式输出结果    指定了非法的输出格式 | | | | | | |

备注：