

《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 04 与 05 班	姓名	胡鑫、冯宇馨
实验题目	浮点数的表示和计算		
实验时间	2023 年 6 月 1 日	实验地点	DS1410
实验成绩	优秀/良好/中等	实验性质	<input checked="" type="checkbox"/> 验证性 <input checked="" type="checkbox"/> 设计性 <input checked="" type="checkbox"/> 综合性
教师评价： <input checked="" type="checkbox"/> 算法/实验过程正确； <input checked="" type="checkbox"/> 源程序/实验内容提交； <input checked="" type="checkbox"/> 程序结构/实验步骤合理； <input checked="" type="checkbox"/> 实验结果正确； <input checked="" type="checkbox"/> 语法、语义正确； <input checked="" type="checkbox"/> 报告规范； 其他： <div>评价教师: 冯永</div>			
实验目的 (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； (3) 掌握 IEEE 754 中单精度浮点数的表示和计算。			

报告完成时间: 2023 年 6 月 8 日

1 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(GUI 或者字符界面)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乘除运算;
4. 可以使用 80X86 或 MIPS 或 ARM 汇编指令,但是不能使用浮点指令,只能利用整数运算指令来编写软件完成。

2 实验过程或算法(源程序)

```
1 .data
2     show_begin: .asciiiz "\nWelcome to our caculator, which operation do you want
        to perform? \n 1.add caculate \n 2.sub caculate \n 3.multiply \n 4.
        divide \n 5.exit \n Choose a operation:"
3     show_exit: .asciiiz "Thank you for choosing our calculator!BYE!"
4     show_cin1: .asciiiz "Please enter the first number:"
5     show_cin2: .asciiiz "Please enter the second number:"
6     show_wrong: .asciiiz "Operation error, please re_enter\n"
7     show_underflow: .asciiiz "underflow error, please re_enter\n"
8     show_overflow: .asciiiz "overflow error, please re_enter\n"
9     show_div: .asciiiz "divider cannot be zero!\n"
10    show_binary: .asciiiz "\nBinary:\n"
11    show_hex: .asciiiz "\nHexdecimal:\n"
12 .text
13
14 #t4 ,t5 ,t6 :进制的符号, 阶数, 尾数
15 main:
16     #开始, 提示用户选择操作
17     la $a0, show_begin
18     li $v0, 4
19     syscall
20     li $v0, 5
21     syscall
22     move $t0, $v0
23     #选择5则退出
24     beq $t0, 5, exit
25     #选择1, 2, 3, 4则进入计算函数
26     jal calutation
27     j main
28
29 calutation:
```

```

30  addi  $sp,  $sp,  -32
31  sw   $ra,  20($sp)
32  sw   $fp,  16($sp)
33  addiu $fp,  $sp,  28
34  sw   $a0,  0($fp)
35
36  #输入第一个浮点数
37  la   $a0,  show_cin1
38  li   $v0,  4
39  syscall
40  li   $v0,  6
41  syscall
42
43  mfc1  $a1,  $f0
44  #取$a1的符号位$s0
45  srl   $s0,  $a1,  31
46  #取$a1的阶码$s1
47  sll   $s1,  $a1,  1
48  srl   $s1,  $s1,  24
49  #取$a1的尾数$s2
50  sll   $s2,  $a1,  9
51  srl   $s2,  $s2,  9
52  #补全隐藏1
53  addi  $s2,  $s2,  0x00800000
54
55  #输入第二个浮点数
56  la   $a0,  show_cin2
57  li   $v0,  4
58  syscall
59  li   $v0,  6
60  syscall
61  mfc1  $a2,  $f0
62
63  #取$a2的符号位$s3
64  srl   $s3,  $a2,  31
65  #取$a2的阶码$s4
66  sll   $s4,  $a2,  1
67  srl   $s4,  $s4,  24
68  #取$a1的尾数$s5
69  sll   $s5,  $a2,  9
70  srl   $s5,  $s5,  9
71  #补全隐藏1
72  addi  $s5,  $s5,  0x00800000
73
74  #加法运算
75  beq   $t0,  1,  add_caculate
76  #减法运算
77  beq   $t0,  2,  sub_caculate
78  #乘法运算

```

```

79     beq  $t0,3, mutiply_caculate
80     #除法运算
81     beq  $t0,4, divide_caculate
82
83 add_caculate:
84 #对阶
85     sub  $t1, $s1, $s4
86     #小数对大数阶
87     bltz $t1, exp1
88     bgtz $t1, exp2
89     beqz $t1, add_sub
90
91 exp1:
92     addi $s1, $s1, 1
93     srl  $s2, $s2, 1
94     j    add_caculate
95
96 exp2:
97     addi $s4, $s4, 1
98     srl  $s5, $s5, 1
99     j    add_caculate
100
101 #判断同号或异号相加
102 add_sub:
103     sub  $t1, $s0, $s3
104     #同号加
105     beqz $t1, add_sum
106     #异号减
107     j    sub_sum
108
109 sub_sum:
110     #比较尾数绝对值大小
111     sub  $t2, $s2, $s5
112     bgtz $t2, a_b
113     bltz $t2, b_a
114     #相等输出0
115     j    cout0
116
117 a_b:
118     #尾数过小或过大
119     blt  $t2, 0x00800000, a_bmove
120     bge  $t2, 0x01000000, a_bmove2
121     j    addsub_result
122
123 a_bmove:
124     sll  $t2, $t2, 1
125     subi $s1, $s1, 1
126     blt  $t2, 0x00800000, a_bmove
127     j    addsub_result

```

```

128
129 a_bmove2:
130     srl    $t2,  $t2,  1
131     add    $s1,  $s1,  1
132     bge    $t2,  0x01000000, a_bmove2
133     j      addsub_result
134
135 b_a:
136     sub    $t2,  $s5,  $s2
137     xori   $s0,  $s0,  0x00000001
138     j      a_b
139
140 add_sum:
141     #尾数相加
142     add    $t2,  $s2,  $s5
143     sge    $t3,  $t2,  0x01000000
144     bgtz   $t3,  yichu
145     j      addsub_result
146
147 #溢出处理
148 yichu:
149     srl    $t2,  $t2,  1
150     add    $s1,  $s1,  1
151     sge    $t3,  $t2,  0x01000000
152     bgtz   $t3,  yichu
153     j      addsub_result
154
155 addsub_result:
156     blt    $s1,  0,    under_flow
157     bgt    $s1,  255,  over_flow
158     sll    $s0,  $s0,  31
159     sll    $s1,  $s1,  23
160     sll    $t2,  $t2,  9
161     srl    $t2,  $t2,  9
162     add    $s1,  $s1,  $t2
163     add    $s0,  $s0,  $s1
164     mtc1   $s0,  $f12
165     li     $v0,  2
166     syscall
167     move   $t4,  $s0
168     j      result
169
170 sub_caculate:
171     xori   $s3,  $s3,  0x00000001
172     j      add_caculate
173
174 mutiply_caculate:
175     #指数是否为零
176     beqz   $s1,  zeroexp

```

```

177     beqz $s4, zeroexptwo
178     j    multnozero
179
180 zeroexp:
181     beq  $s2, 0x800000, multZero
182     beqz $s4, zeroexptwo
183     j    multnozero
184
185 zeroexptwo:
186     beq  $s5, 0x800000, multZero
187     j    multnozero
188
189 #有0就输出0
190 multZero:
191     li   $t4, 0
192     li   $t5, 0
193     li   $t6, 0
194     j    cout0
195
196 #无零乘法运算
197 multnozero:
198     #指数相加
199     add  $t5, $s1, $s4
200     li   $t7, 127
201     sub  $t5, $t5, $t7
202     #尾数相乘
203     mult $s2, $s5
204     mfhi $t6    #高32位
205     mflo $t7    #低32位
206     sll  $t6, $t6, 9
207     srl  $t7, $t7, 23
208     or   $t6, $t6, $t7
209     #规格化
210     srl  $t7, $t6, 24
211     beqz $t7, normalized
212     srl  $t6, $t6, 1
213     addi $t5, $t5, 1
214
215 normalized:
216     xor  $t4, $s0, $s3
217     #溢出
218     sgt  $t7, $t5, $zero
219     beqz $t7, under_flow
220     slti $t7, $t5, 255
221     beqz $t7, over_flow
222     j    mult_result
223
224 mult_result:
225     sll  $t4, $t4, 31

```

```

226     sll    $t5,  $t5,  23
227     sll    $t6,  $t6,  9
228     srl    $t6,  $t6,  9
229     add    $t5,  $t5,  $t6
230     add    $t4,  $t4,  $t5
231     mtc1   $t4,  $f12
232     li     $v0,  2
233     syscall
234     j      result
235
236 #除法
237 #t0:选择操作; $t1:对阶操作 ; $t2:尾数加减 ; $t3:判断是否溢出
238 #t4 ,t5 ,t6是结果的符号, 阶数, 尾数
239 divide_caculate:
240     #被除数是否为0
241     beqz   $s1,  divzeroexp
242     j      divnozero
243
244 divzeroexp:
245     beq    $s2,  0x800000,  divZero
246     j      divnozero
247
248 divZero:
249     li     $t4,  0
250     li     $t5,  0
251     li     $t6,  0
252     j      cout0
253
254 divnozero:
255     #除数是否为0
256     bnez   $s4,  divoperate
257     bne    $s5,  0x800000,  divoperate
258     la     $a0,  show_div
259     li     $v0,  4
260     syscall
261     j      main
262
263 divoperate:
264     #指数相减
265     sub    $t5,  $s1,  $s4
266     addi   $t5,  $t5,  127
267     #符号
268     xor    $t4,  $s0,  $s3
269     #尾数相除
270     div    $s2,  $s5
271     mflo   $t6 #商
272     mfhi   $t7 #余数
273     beqz   $t6,  div_result
274     li     $t8,  1

```

```

275
276 #商右移
277 divloop1:
278     srlv $t9, $t6, $t8
279     bne $t9, $zero, divloop1
280     li $t9, 1
281     sub $t8, $t8, $t9
282     #指数加
283     add $t5, $t5, $t8
284
285     #溢出
286     slti $t2, $t5, 0
287     beq $t2, 1, under_flow
288     li $t2, 255
289     slt $t2, $t2, $t5
290     beq $t2, 1, over_flow
291
292     li $t3, 23
293     sub $t3, $t3, $t8
294     #计数器
295     li $t1, 0
296
297 divloop2:
298     sll $t7, $t7, 1
299     div $t7, $s5
300     mflo $k0
301     mfhi $t7
302     #尾数
303     sll $t6, $t6, 1
304     add $t6, $t6, $k0
305     addi $t1, $t1, 1
306     beq $t1, $t3, div_result
307     beqz $t7, div_comp_dec
308     j divloop2
309
310 div_comp_dec:
311     sub $t1, $t3, $t1
312     sllv $t6, $t6, $t1
313
314 div_result:
315     sll $t4, $t4, 31
316     sll $t5, $t5, 23
317     sll $t6, $t6, 9
318     srl $t6, $t6, 9
319     add $t5, $t5, $t6
320     add $t4, $t4, $t5
321     mtc1 $t4, $f12
322     li $v0, 2
323     syscall

```



```

324     j result
325
326 #下溢
327 under_flow :
328     la $a0, show_underflow
329     li $v0, 4
330     syscall
331     j main
332
333 #上溢:
334 over_flow :
335     la $a0, show_overflow
336     li $v0, 4
337     syscall
338     j main
339
340 #输出0
341 cout0:
342     li $a0, 0
343     li $v0, 1
344     syscall
345     j main
346
347 #打印输出结果
348 result:
349     #”Binary:”
350     la $a0, show_binary
351     li $v0, 4
352     syscall
353
354     binary:
355     # $t7 用于计数
356     addu $k0, $t4, $zero
357     addi $t7, $0, 32
358     addi $t8, $0, 0x80000000
359     addi $t9, $0, 0
360     binary_output:
361     subi $t7, $t7, 1
362     and $t9, $k0, $t8
363     srl $t8, $t8, 1
364     srlv $t9, $t9, $t7
365     add $a0, $t9, $0
366     #循环输出, 一次一位
367     li $v0, 1
368     syscall
369     beq $t7, $0, hexbegin
370     j binary_output
371
372 #十六进制输出

```

```

373 hexbegin:
374     #”Hex:”
375     la $a0, show_hex
376     li $v0, 4
377     syscall
378
379     hex:
380     li $t7, 0
381     addi $t7, $zero, 8
382     add $t9, $t4, $zero
383     add $t8, $t4, $zero
384
385     hex_output:
386     beqz $t7, return
387     addi $t7, $t7, -1
388     srl $t8, $t9, 28
389     sll $t9, $t9, 4
390     bgt $t8, 9, hex_char
391     li $v0, 1
392     addi $a0, $t8, 0
393     syscall
394     j hex_output
395
396     hex_char:
397     addi $t8, $t8, 55
398     li $v0, 11
399     addi $a0, $t8, 0
400     syscall
401     j hex_output
402
403     return:
404     lw $t1, 28($sp)
405     lw $t6, 24($sp)
406     lw $ra, 20($sp)
407     lw $fp, 16($sp)
408     addiu $sp, $sp, 32
409     jr $ra
410     #退出
411     exit:
412     la $a0, show_exit
413     li $v0, 4
414     syscall

```

3 实验结果及分析和(或)源程序调试过程

3.1 选择需要进行的操作

```
Welcome to our caculator,which operation do you want to perform?  
1.add caculate  
2.sub caculate  
3.multiply  
4.divide  
5.exit  
Choose a operation:
```

3.2 输入要进行计算的两个数值

```
Choose a operation:1  
Please enter the first number:1.5  
Please enter the second number:2.6
```

3.3 加法操作及其结果

输入 1,选择加法操作 (如上图所示)。

```
4.1  
Binary:  
0100000010000011001100110011  
Hexdecimal:  
40833333  
Welcome to our caculator,which operation do you want to perform?  
1.add caculate  
2.sub caculate  
3.multiply  
4.divide  
5.exit  
Choose a operation:
```

输出的答案有十进制,二进制与十六进制,输出的二进制和十六进制是计算结果在内存中以 IEEE 754 单精度方式表示的形式。同时我们还实现了科学计数法的输出方式,但由于重复代码部分,源码部分就未粘贴。下图为我们科学计数法的输出:

```
4.1
Binary:
1.00000110011001100110011*2^2
Hexdecimal:
4.199998*16^0

Welcome to our caculator,which operation do you want to perform?
1.add caculate
```

由二进制,十六进制及其科学计数法可知,此运算正确。

3.4 减法操作及其结果

输入 2,选择减法操作。

```
Choose a operation:2
Please enter the first number:2.8
Please enter the second number:1.3
1.5
Binary:
00111111110000000000000000000000
Hexdecimal:
3FC00000
```

科学计数法输出:

```
1.5
Binary:
1.100000000000000000000000*2^0
Hexdecimal:
1.800000*16^0
```

由图可知,答案正确。

3.5 乘法操作及其结果

输入 3, 选择乘法操作。

```
Choose a operation:3
Please enter the first number:1.6
Please enter the second number:3
4.7999997
Binary:
01000000100110011001100110011001
Hexdecimal:
40999999
```

科学计数法输出:

```
4.7999997
Binary:
1.00110011001100110011001*2^2
Hexdecimal:
4.CCCCC8*16^0
```

由图可知, 答案正确。

3.6 除法操作及其结果

输入 4, 选择除法操作。

```
Choose a operation:4
Please enter the first number:5
Please enter the second number:2
2.5
Binary:
01000000001000000000000000000000
Hexdecimal:
40200000
```

科学计数法输出:

```
2.5
Binary:
1.010000000000000000000000*2^1
Hexdecimal:
2.800000*16^0
```

由图可知, 答案正确。

3.7 退出程序

```
Choose a operation:5
Thank you for choosing our calculator!BYE!
— program is finished running (dropped off bottom) —
```

4 遇到的问题及解决办法

在完成此项目的过程中, 我们也遇到了很多困难, 从最开始的如何输出语句输入指令等最基础的功能实现, 到后期对浮点数运算的思考, 我们在完成十进制的输出时, 我们得到了结果的符号位, 指数, 尾数, 因为 IEEE 的存储需要隐藏 1, 刚开始时忘记此步操作, 一直出现问题, 后面通过左移 9 位, 将第 24 位的 1 移除, 又右移, 达到效果, 成功输出。