

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH



## ASSIGNMENT 1

---

# NHÓM L07-12 - ĐỀ 1: CỘNG TRỪ HAI SỐ THỰC CHUẨN VỚI ĐỘ CHÍNH XÁC ĐƠN

---

GVHD: Nguyễn Xuân Minh  
Đinh Đức Anh Vũ

SV: Võ Hoàng Hải Nam    MSSV: 1810340  
Phan Quốc Long    MSSV: 1810299  
Huỳnh Thị Uyên    MSSV: 1810648

TP. HỒ CHÍ MINH, THÁNG 12/2019



## Mục lục

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Giải thuật và các bước thực hiện code</b> | <b>2</b>  |
| <b>2</b> | <b>Tính thời gian chạy của chương trình</b>  | <b>3</b>  |
| <b>3</b> | <b>Phân code của nhóm</b>                    | <b>5</b>  |
| <b>4</b> | <b>Ví dụ cụ thể</b>                          | <b>10</b> |
| 4.1      | Ví dụ 1 . . . . .                            | 10        |
| 4.2      | Ví dụ 2 . . . . .                            | 10        |
| 4.3      | Ví dụ 3 . . . . .                            | 11        |
| 4.4      | Ví dụ 4 . . . . .                            | 11        |
| <b>5</b> | <b>Tài liệu tham khảo</b>                    | <b>12</b> |

## Đề 1: Cộng, trừ 2 số thực.

Cho 2 số thực dạng chuẩn (Standard Floating Point IEEE 754) A và B với độ chính xác đơn (32 bit). Sử dụng hợp ngữ assembly MIPS, viết thủ tục cộng (trừ) hai số A, B.

# 1 Giải thuật và các bước thực hiện code

## 1. Bước 1: Tách số thực thành các thành phần

- Số thực chính xác đơn gồm | 1 bit dấu | 8 bit số mũ | 23 bit phân số |.
- Lấy giá trị từng thành phần của Num1 và Num2 và lưu vào các thanh ghi thường.
- Gán 1 cho bit thứ 24 của mỗi phần định trị để được dạng chuẩn.

## 2. Bước 2: So sánh phần mũ và dịch phần phân số

- So sánh 2 phần mũ, nếu số mũ Num1 < số mũ Num2, dịch phải 1 bit phần định trị Num1 và đồng thời cộng số mũ cho 1, lặp lại tới khi bằng số mũ Num2 và ngược lại.
- Sau đó xét Num1 và Num2, nếu phần bit dấu bằng 1 (âm) thì đổi dấu phần định trị của số đó.

## 3. Bước 3: Cộng/trừ hai phần định trị và chuẩn hóa

- Đối với phép cộng, ta cộng hai phần định trị. Sau đó tiến hành quy trình kiểm tra dấu kết quả cộng.
- Nếu kết quả cộng hai phần định trị âm thì ta đổi dấu kết quả và lưu lại bit dấu.
- Tiếp theo, xét kết quả có tràn quá 24 bit hay không. Nếu có thì ta dịch phải 1 bit, lấy 24 bit thấp làm kết quả phần định trị và đồng thời tăng số mũ lên 1. Ngược lại, tiến hành chuẩn hoá để bảo đảm dạng chuẩn IEEE 754.
- Sau cùng, thực hiện ghép kết quả để được kết quả hoàn chỉnh và quay lại đầu bước 3 để thực hiện tương tự với trừ 2 phần định trị.
- Cuối cùng xuất kết quả ra màn hình.

## 2 Tính thời gian chạy của chương trình

Chạy trên máy tính kiến trúc MIPS có tần số 3.4 GHz theo mô hình đơn chu kỳ (Single cycle).

### 1. TH1: Số thực dương cộng / trừ số thực dương

Ta chọn:  $a = 7.995$ ,  $b = 3.115$ . Kết quả thu được là:

$a + b = 11.11$ ,  $a - b = 4.88$ . (Xem phần ví dụ 1)

Tổng số các lệnh MIPS là 115, bao gồm:

- R-type: 48
- I-type: 59
- J-type: 7

#### Sử dụng mô hình đơn chu kỳ (Single cycle)

Với mô hình đơn chu kỳ, tất cả các lệnh đều được thực hiện 1 chu kỳ.

Tổng số chu kỳ = 115

$$\text{Thời gian thực hiện} = \frac{CPUclockcycles}{clockrate} = \frac{115}{3.4 \times 10^9} = 33.8235(ns)$$

### 2. TH2: Số thực dương cộng / trừ số thực âm

Ta chọn:  $a = 23.45678$ ,  $b = -98.765432$ . Kết quả thu được:

$a + b = -75.308655$ ;  $a - b = 122.22221$  (Xem phần ví dụ 2).

Tổng số các lệnh MIPS là 124, bao gồm:

- R-type: 52
- I-type: 64
- J-type: 7

#### Sử dụng mô hình đơn chu kỳ (Single cycle)

Với mô hình đơn chu kỳ, tất cả các lệnh đều được thực hiện 1 chu kỳ.

Tổng số chu kỳ = 124

$$\text{Thời gian thực hiện} = \frac{CPUclockcycles}{clockrate} = \frac{124}{3.4 \times 10^9} = 36.4706(ns)$$

### 3. TH3: Số thực âm / trừ số thực dương

Ta chọn:  $a = -12.34$ ,  $b = 23.12$ . Kết quả thu được là:  
 $a + b = 10.780001$ ,  $a - b = -35.46$ . (Xem phần ví dụ 3)  
Tổng số các lệnh MIPS là 128, bao gồm:

- R-type: 54
- I-type: 65
- J-type: 8

#### Sử dụng mô hình đơn chu kỳ (Single cycle)

Với mô hình đơn chu kỳ, tất cả các lệnh đều được thực hiện 1 chu kỳ.

Tổng số chu kỳ = 128

$$\text{Thời gian thực hiện} = \frac{CPUclockcycles}{clockrate} = \frac{128}{3.4 \times 10^9} = 37.6471(ns)$$

### 4. TH4: Số thực âm / trừ số thực âm

Ta chọn:  $a = -12.34$ ,  $b = -6.78$ . Kết quả thu được là:  
 $a + b = -19.119999$ ;  $a - b = -5.5600004$ . (Xem phần ví dụ 4)  
Tổng số các lệnh MIPS là 133, bao gồm:

- R-type: 57
- I-type: 67
- J-type: 8

#### Sử dụng mô hình đơn chu kỳ (Single cycle)

Với mô hình đơn chu kỳ, tất cả các lệnh đều được thực hiện 1 chu kỳ.

Tổng số chu kỳ = 133

$$\text{Thời gian thực hiện} = \frac{CPUclockcycles}{clockrate} = \frac{133}{3.4 \times 10^9} = 39.1176(ns)$$

### 3 Phần code của nhóm

```
1 #####
2 #####-----#####
3 #####          CODE CONG VA TRU 2 SO THUC CHINH XAC DON          #####
4 #####          Nhom L07_12          #####
5 #####-----#####
6 #####
7
8 # Luu 2 so thuc chinh xac don vao vung .data
9 .data
10     # Cac chuoi xuat ra console
11     str1:  .ascii "Moi nhap vao a: "
12     str2:  .ascii "Moi nhap vao b: "
13     str3:  .ascii "Ket qua a + b: "
14     str4:  .ascii "\nKet qua a - b: "
15
16 # Bat dau vung .text
17 .text
18 # Bat dau chay o ham main
19     j main
20
21
22 #-----#
23 #          B1: TACH SO THUC THANH CAC THANH PHAN          #
24 #-----#
25 # So thuc chinh xac don: |1 bit dau|8 bit so mu|23 bit phan so| #
26 # Ta se lay cac bit nay ra bang cach andi hoac ori Num1/Num2 roi #
27 # sao chep gia tri trong do          #
28 # De de dang tinh toan ta se luu Num1/Num2 vao cac thanh ghi    #
29 # binh thuong ma khong su dung thanh ghi so thuc          #
30 #-----#
31
32 # -----$t0 = a; $t1 = b -----#
33 # $s0 = t0[31] ; $s1 = t0[30:23]; $s2 = [1:t0[22:0]] -----#
34 # $s3 = t1[31] ; $s4 = t1[30:23]; $s5 = [1:t0[22:0]] -----#
35
36
37 #-----Load cac gia tri cua Num1-----#
38 #-----(bit dau vao $s0, phan mu vao $s1, phan phan so vao $s2)----#
39
40
41 Func:
42     # Lay gia tri 1 bit dau luu vao $s0
43     andi $s0,$t0,0x80000000
44     # Lay gia tri 8 bit so mu luu vao $s1
45     andi $s1,$t0,0x7f800000
46     # Lay gia tri 23 bit phan so luu vao $s2
```

```
47     andi    $s2,$t0,0x007fffff
48     # them 1 vao dau de thu duoc phan dinh tri 24bit
49     ori     $s2,$s2,0x00800000
50 #-----end-----#
51
52 #-----Load cac gia tri cua Num2-----#
53 #----(bit dau vao $s3, phan mu vao $s4, phan phan so vao $s5)----#
54
55     # Lay gia tri 1 bit dau luu vao $s3
56     andi    $s3,$t1,0x80000000
57     # Lay gia tri 8 bit so mu luu vao $s4
58     andi    $s4,$t1,0x7f800000
59     # Lay gia tri 23 bit phan so luu vao $s5
60     andi    $s5,$t1,0x007fffff
61     # them 1 vao dau de thu duoc phan dinh tri 24 bit
62     ori     $s5,$s5,0x00800000
63 #-----END B1-----#
64
65
66 #-----#
67 #           B2: SO SANH MU VA DICH PHAN SO           #
68 #-----#
69 # Sau khi thuc hien doan code tren:                  #
70 # Num1: | $s0 | $s1 | $s2 |                          #
71 # Num2: | $s3 | $s4 | $s5 |                          #
72 # Thuc hien so sanh phan so mu, dich phan phan so cho phu hop #
73 # Cong hai phan phan so, chuan hoa                  #
74 #-----#
75
76 #-----So sanh phan mu va dich phan dinh tri-----#
77
78     # So sanh mu, neu bang nhau thi cong phan dinh tri ngay
79     beq     $s1,$s4,XetNum1
80     slt     $t7,$s1,$s4
81     # Neu so mu Num1 < so mu Num2 thi dich phan dinh tri cua Num1
82     beq     $t7,1,DichNum1
83     # Dich Num2
84 DichNum2:
85     beq     $s1,$s4,XetNum1
86     # Dich phai phan dinh tri cua Num2
87     srl     $s5,$s5,1
88     # phan mu + 1 <=> bit[30:23]+1 -> tuong ung vi tri phan mu
89     addi    $s4,$s4,0x00800000
90     j       DichNum2
91
92 DichNum1:
93     beq     $s1,$s4,XetNum1
94     # Dich phai phan dinh tri cua Num2
```



```
95     srl    $s2,$s2,1
96     #phan mu + 1 <=> bit[30:23]+1 -> tuong ung vi tri phan mu
97     addi   $s1,$s1,0x00800000
98     j      DichNum1
99
100  XetNum1:    # neu s0 = 1 (num1<0) -> doi dau phan dinh tri num1
101     beq    $s0,0,XetNum2
102     sub    $s2,$zero,$s2
103  XetNum2:
104     # neu s3 = 1 (num2<0) -> doi dau phan dinh tri num2
105     beq    $s3,0,Congdinhtri
106     sub    $s5,$zero,$s5
107
108  #-----end-----#
109
110
111  #-----Cong hai phan dinh tri-----#
112  #----- lay 24 bit thap luu vao $s6, phan mu luu vao $t8 -----#
113  # Tham so $a0 = $s2 + $s5// $a0 = $s2- $s5 (phan dinh tri)-----#
114  # Tra ve kq $v0 -----#
115
116  Congdinhtri:
117     # Tong/Hieu hai phan dinh tri: $a0 = $s2+$s5 // $s2- $s5
118     # Bit dau luu vao $t2
119     # $t4: giu so mu de thuc hien tang giam
120     # $t8 = Tong  $t9 = Hieu
121
122     add    $a0,$s2,$s5
123     # Thuc hien phep cong -> KQ = $v0
124     jal    Xetdaudinhtri
125     #Luu kq vao $t8
126     add    $t8,$v0,$zero
127
128     # Phep tru 2 dinh tri
129     sub    $a0,$s2,$s5
130     jal    Xetdaudinhtri
131     add    $t9,$v0,$zero
132     j      printResult
133
134  Xetdaudinhtri:
135     # Neu dinh tri < 0 -> $t2 = 1
136     slt    $t2,$a0,$zero
137
138     # Neu $t2 = 1 -> $a0 < 0 -> $t2 =0x80000000 (bit dau)
139     # va doi dau a0 vi a0 > 0
140     # $t2 = 0 -> bit dau = $t2 = 0
141     beq    $t2,0,KTraDinhTri
142     addi   $t2,$zero,0x80000000
143     sub    $a0,$zero,$a0
```



```
144
145 KTraDinhTri:
146     # Lay ra bit cao nhat cua phan dinh tri -> t7 (overflow)
147     srl    $t7,$a0,24
148     # t7 = 1 phan dinh tri 25 bit -> dich phai 24 bit,
149     # lay 24 bit thap lam phan dinh tri
150     # dong thoi: mu-> mu + 1
151     add    $t4,$zero,$s1
152     # t7 = 0 tien hanh vong lap -> chuan tac
153     beq    $t7,0,Chuantac
154     # phan mu + 1 <=> bit[30:23] + 1
155     # -> tuong ung voi vi tri phan mu cua so thuc
156     addi   $t4,$t4,0x00800000
157     # Dich phan dinh tri qua phai 1 bit
158     srl    $a0,$a0,1
159     j      GhepKQ
160
161 Chuantac:
162     srl    $t7,$a0,23
163     beq    $t7,1,GhepKQ
164     # Dich trai phan phan so cua ket qua
165     sll    $a0,$a0,1
166     # phan mu - 1 <=> bit[30:23] - 1
167     # -> tuong ung voi vi tri phan mu cua so thuc
168     subi   $t4,$t4,0x00800000
169     # Kiem tra tiep tục
170     j      Chuantac
171 GhepKQ:
172     # Lay 23 bit thap -> dinh tri cho ket qua
173     andi   $v0,$a0,0x007fffff
174     # Ghep bit dau ( t2) vao thanh a0
175     or     $v0,$v0,$t2
176     # ghep 8 bit mu (s1) vao thanh a0 ghi ket qua
177     or     $v0,$v0,$t4
178     jr     $ra
179
180 #-----END B3-----#
181
182
183 #-----#
184 #                                     HAM MAIN                                #
185 #-----#
186 # Input: hai so thuc chinh xac don                                           #
187 # Output: ket qua cong va tru                                                 #
188 #-----#
189 main:
190     # Nhap va gia tri a, luu vao Num1
191     addi   $a0,$zero,0x10010000
192     addi   $v0,$zero,4
```

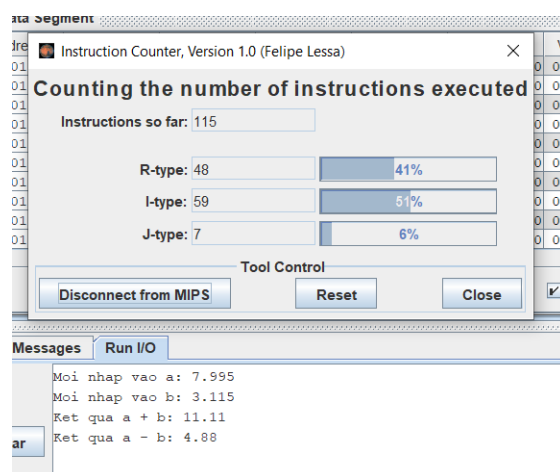
```
193     syscall
194
195     addi    $v0,$zero,6
196     syscall
197     mfc1    $t0,$f0
198
199     # Nhập và gia tri b, lưu vào Num2
200     addi    $a0,$zero,0x10010011
201     addi    $v0,$zero,4
202     syscall
203
204     addi    $v0,$zero,6
205     syscall
206     mfc1    $t1,$f0
207     j Func
208
209 printResult:
210     # Load vào thanh ghi $f12 và gọi syscall in ra console
211     addi    $a0,$zero,0x10010022
212     addi    $v0,$zero,4
213     syscall
214
215     mtc1    $t8,$f12
216     addi    $v0,$zero,2
217     syscall
218
219     addi    $a0,$zero,0x10010032
220     addi    $v0,$zero,4
221     syscall
222
223     mtc1    $t9,$f12
224     addi    $v0,$zero,2
225     syscall
226
227 #####
228 #####-----#####
229 #####                                END PROGRAM                                #####
230 #####-----#####
231 #####
232
```

Listing 1: MIPS code

## 4 Ví dụ cụ thể

### 4.1 Ví dụ 1

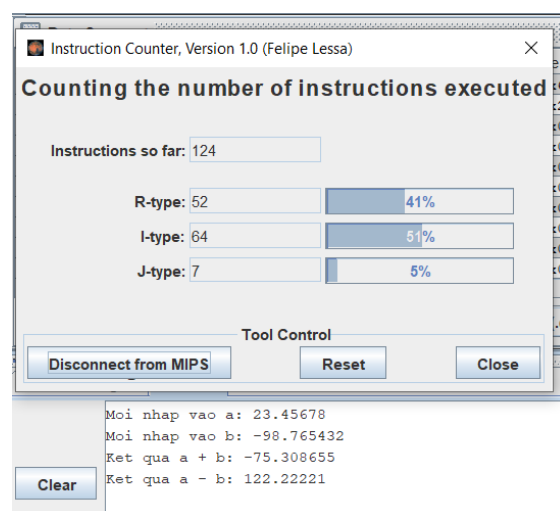
$a = 7.995$  và  $b = 3.115$



Hình 1: Ví dụ 1

### 4.2 Ví dụ 2

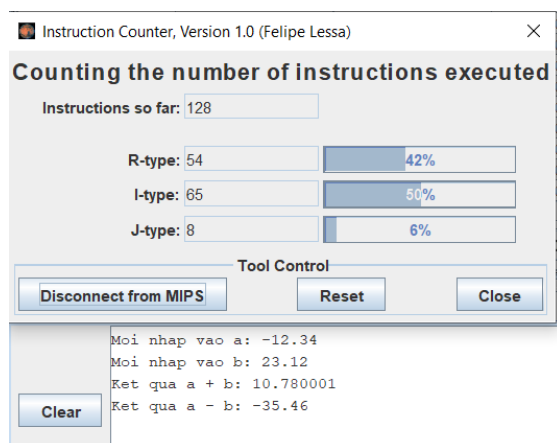
$a = 23.45678$  và  $b = -98.765432$



Hình 2: Ví dụ 2

### 4.3 Ví dụ 3

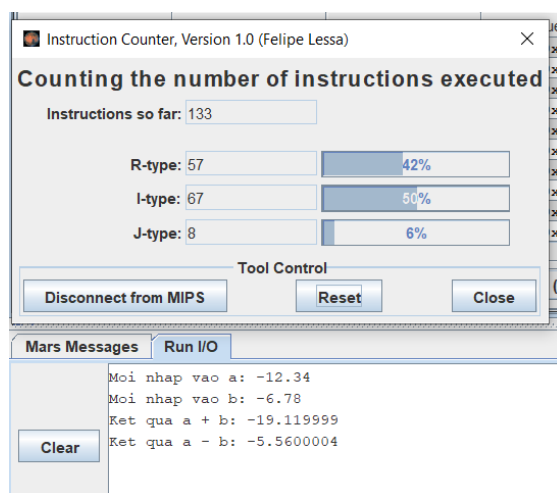
$a = -12.34$  và  $b = 23.12$



Hình 3: Ví dụ 3

### 4.4 Ví dụ 4

$a = -12.34$  và  $b = -6.78$



Hình 4: Ví dụ 4



## 5 Tài liệu tham khảo

1. David A. Patterson, John L. Hennessy (2014). Computer Organization and Design: The Hardware/ Software Interface, Fifth Edition.
2. Phạm Quốc Cường (2017). Kiến trúc máy tính, NXB Đại học quốc gia TP.HCM.