

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**



**BÁO CÁO BÀI TẬP LỚN**  
**MÔN KIẾN TRÚC MÁY TÍNH**

Giảng viên hướng dẫn: Nguyễn Xuân Minh

Lớp L02\_Nhóm 34

Sinh viên:    Phạm Nguyên Hải                   1913261

                  Phan Nguyễn Xuân Lộc           2113971

*Học kì 222, năm học 2022-2023*

## MỤC LỤC

1. Đề bài .....	3
2. Yêu cầu .....	3
3. Ý tưởng thực hiện .....	3
4. Trình tự thực hiện .....	3
5. Thống kê số lệnh, loại lệnh (instruction type) và thời gian chạy .....	4
5.1 Ví dụ 1 .....	4
5.2 Ví dụ 2 .....	5
5.3 Ví dụ 3 .....	5
5.4 Ví dụ 4 .....	6
5.5 Ví dụ 5 .....	6
5.6 Ví dụ 6 .....	7
Tài liệu tham khảo .....	7

## 1. Đề bài

Chia 2 số thực chuẩn IEEE 754 chính xác đơn.

Viết chương trình thực hiện phép chia 2 số thực chuẩn IEEE 754 chính xác đơn mà không dùng các lệnh tính toán số thực của MIPS.

## 2. Yêu cầu

- ❖ Sử dụng tập lệnh MIPS để hiện thực các thủ tục bên dưới.
- ❖ Thống kê số lệnh, loại lệnh (instruction type) của mỗi chương trình.
- ❖ Tính thời gian chạy của chương trình.
- ❖ Code
  - Code style phải rõ ràng, có comment.
  - Phải có gọi hàm. Truyền tham số và trả kết quả khi gọi hàm theo quy ước của thanh ghi (\$A0 chứa tham số, \$V0 hoặc \$f0 chứa giá trị trả về).
  - In kết quả ra màn hình để kiểm tra.
- ❖ Báo cáo
  - Trong báo cáo cần nêu rõ các dữ liệu mẫu dùng để kiểm tra.
  - Báo cáo gồm có file báo cáo (không source code) định dạng .PDF (Bc\_nhom##.pdf) và phần mã nguồn đi kèm (Mn\_nhom##.asm).

## 3. Ý tưởng thực hiện

Biểu diễn một số thực theo định dạng IEEE 754 sẽ gồm có 3 phần:

- "S" biểu diễn dấu (sign) là bit thứ 31 (1 bit).
- "Phần mũ" (exponent) là bit 23-30 (8 bit)
- "Phần phân số" (fraction) là bit 0-22 (23 bit)

Sau đó ta tính toán trên từng phần.

## 4. Trình tự thực hiện

Chia 2 số thực bất kỳ ta thực hiện theo trình tự sau:

- Nhập và lưu giá trị của 2 số thực A và B. Sau đó load giá trị vào các thanh ghi.
- Kiểm tra 3 trường hợp đặc biệt:
  - $A = 0$  và  $B = 0$ . Khi đó, ta không thể thực hiện phép chia. Nên  $A/B = \text{NaN}$ .
  - $A = 0$  và  $B \neq 0$ . Đây là phép chia của 0 cho một số thực bất kỳ vậy kết quả:  $A/B = 0$ .
  - $A \neq 0$  và  $B = 0$ . Đây là phép chia của một số thực khác 0 cho 0. Vậy kết quả sẽ là  $A/B = \text{Infinity}$ .
- Nếu không phải 1 trong 3 trường hợp trên tức là  $A \neq 0$  và  $B \neq 0$  thì ta sẽ đưa cả 2 số thực về dạng IEEE 754 như sau:
  - Dịch A và B sang phải 31 bit ta sẽ xác định được bit dấu S. Nếu số thực dương thì  $S = 1$  còn số thực âm thì  $S = 0$ .
  - Dịch trái A và B 1 bit để bỏ bit dấu. Sau đó dịch phải 24 bit. Ta sẽ thu được 8 bit của phần mũ E.

- Dịch trái A và B 9 bit để bỏ đi bit dấu và bit của phần mũ, sau đó dịch sang phải 9 bit. Ta sẽ thu được 23 bit của phần phân số F.
- Sau khi tách ra làm 3 phần. Ta sẽ tính toán trên từng phần như sau:
  - Ta sẽ lấy 2 bit dấu S của A và B dùng phép XOR để xác định dấu của phép chia.
  - Xác định phần mũ bằng cách. Lấy phần mũ của A trừ đi B cộng bias (127). Tức là:  $EA - EB + \text{bias}$ .
  - Ở phần phân số (fraction) thì ta sẽ thêm 1 tại bit thứ 23 bằng cách lấy fraction ori với  $0x00800000$ . Sau đó ta dùng vòng lặp (loop) để lấy ra 24 bit của kết quả phép chia đồng thời kiểm tra xem kết quả có dư hay là không? Nếu có thì cộng kết quả trên thêm 1 (vì nếu dư thì sẽ làm tròn lên). Sau đó ta lại tiếp tục thực hiện andi với  $0x007FFFFFFF$  để bỏ 1 ở bit thứ 23 đi. Cuối cùng ta thu được 23 bit của fraction.
- Sau đó ta tổng hợp 3 phần lại sẽ được kết quả của phép chia 2 số thực.

## 5. Thống kê số lệnh, loại lệnh (instruction type) và thời gian chạy

Sử dụng công cụ Instruction Counter để đếm số lệnh trong chương trình:

Tools → Instruction Counter.

Sử dụng công thức sau để tính thời gian chạy:

$$\text{CPU time} = \frac{\text{CPU clock cycles}}{\text{Clock rate}} = \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock rate}}$$

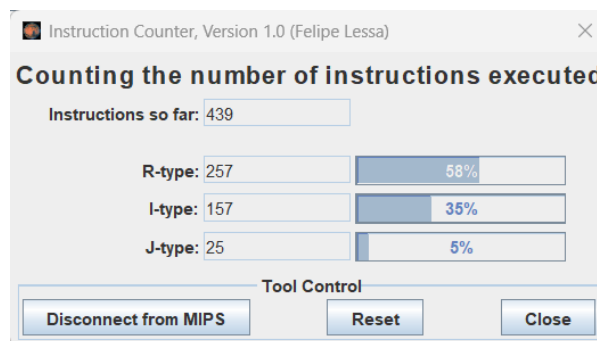
Trong đó:

- CPU time là thời gian xử lý của chương trình (không tính thời gian giao tiếp I/O, thời gian chờ ...).
- CPU Clock cycles: Tổng số chu kỳ thực thi.
- Instruction Count là tổng số lệnh thực thi của chương trình.
- CPI (cycle per instruction) là số chu kỳ thực thi trên một lệnh.
- Clock rate là số chu kỳ thực thi trên một giây hay còn gọi là tần số. Ở bài tập lớn này, ta mặc định Clock rate là 1.8 GHz.

Dưới đây là một số ví dụ cho thấy thời gian chạy khác nhau:

### 5.1 Ví dụ 1

Nhập vào: A=6, B=5. Kết quả: A/B= 1.2 .



Tổng số lệnh: 439

Trong đó:

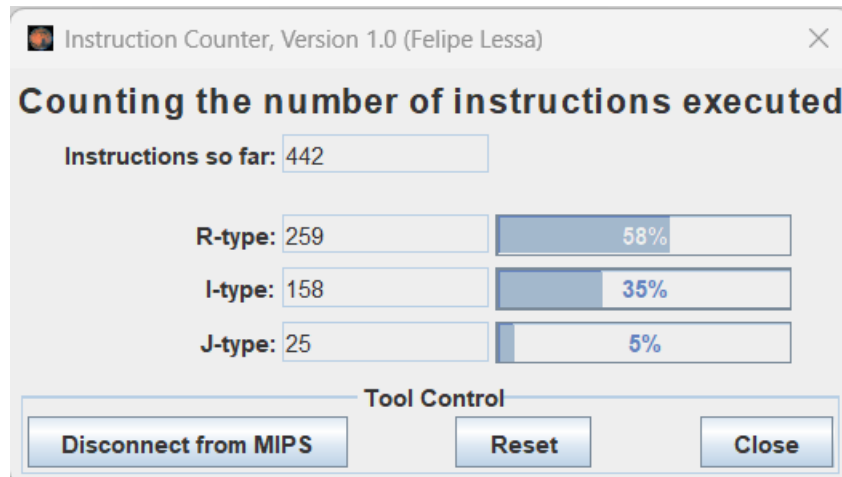
- R-type: 251 lệnh chiếm 58%.
- I-type: 157 lệnh chiếm 35%.
- J-type: 25 lệnh chiếm 5%.

Tính thời gian chạy:

$$\text{CPU time} = \frac{439 \times 1}{1.8 \times 10^9} = 0.2778 \times 10^{-6} \text{ (s)} = 0.2439 \text{ (}\mu\text{s)}$$

## 5.2 Ví dụ 2

Nhập vào: A=9.564, B=3.565. Kết quả: A/B = 2.682749 .



Tổng số lệnh: 442

Trong đó:

- R-type: 259 lệnh chiếm 58%.
- I-type: 158 lệnh chiếm 35%.
- J-type: 25 lệnh chiếm 5%.

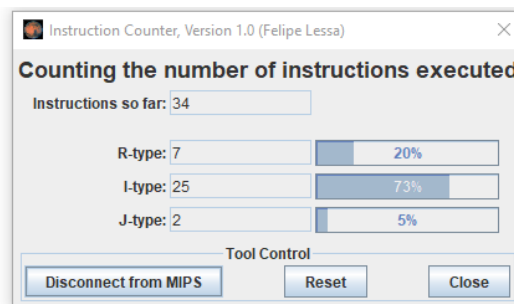
Tính thời gian chạy:

$$\text{CPU time} = \frac{415}{1.8 \times 10^9} = 0,2306 \times 10^{-6} \text{ (s)}$$

## 5.3 Ví dụ 3

Nhập vào: A=3, B=0. Kết quả: A/B = Infinity .

```
Nhap so A : 3
Nhap so B : 0
A / B = Infinity
-- program is finished running --
```



Tổng số lệnh: 34

Trong đó:

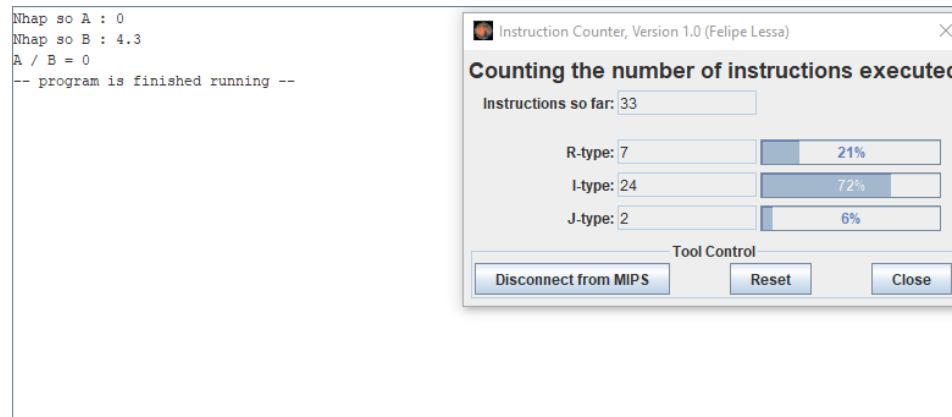
- R-type: 7 lệnh chiếm 20%.
- I-type: 25 lệnh chiếm 73%.
- J-type: 2 lệnh chiếm 5%.

Tính thời gian chạy:

$$\text{CPU time} = \frac{34 \times 1}{1.8 \times 10^9} = 0.0189 \times 10^{-6} \text{ (s)} = 0.0189 \text{ (}\mu\text{s)}$$

## 5.4 Ví dụ 4

Nhập vào: A=0, B=4.3. Kết quả: A/B = 0 .



Tổng số lệnh: 33

Trong đó:

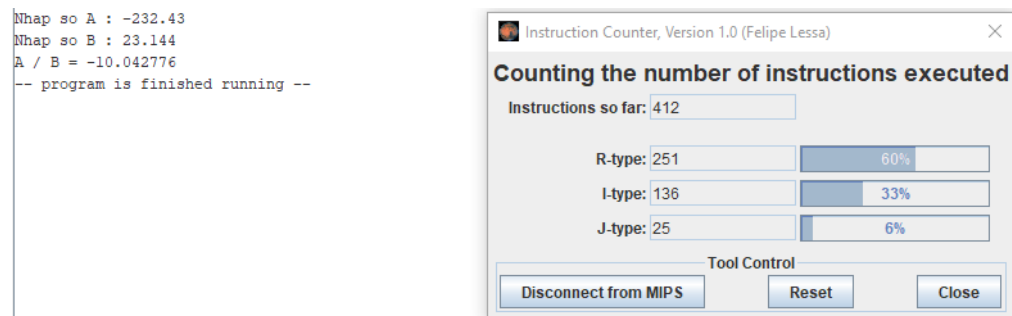
- R-type: 7 lệnh chiếm 21%.
- I-type: 24 lệnh chiếm 72%.
- J-type: 2 lệnh chiếm 6%.

Tính thời gian chạy:

$$\text{CPU time} = \frac{33 \times 1}{1.8 \times 10^9} = 0.0183 \times 10^{-6} \text{ (s)} = 0.0183 \text{ (}\mu\text{s)}$$

## 5.5 Ví dụ 5

Nhập vào: A= -232.43, B=23.144. Kết quả: A/B= -10.042776 .



Tổng số lệnh: 412

Trong đó:

- R-type: 251 lệnh chiếm 60%.
- I-type: 136 lệnh chiếm 33%.
- J-type: 25 lệnh chiếm 6%.

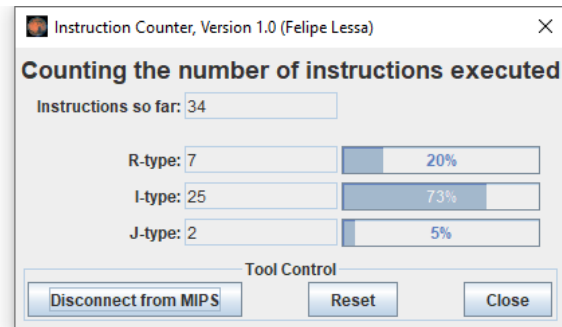
Tính thời gian chạy:

$$\text{CPU time} = \frac{410 \times 1}{1.8 \times 10^9} = 0.2777 \times 10^{-6} \text{ (s)} = 0.2777 \text{ (}\mu\text{s)}$$

## 5.6 Ví dụ 6

Nhập vào: A=0, B=0. Kết quả: A/B= NaN .

```
Nhap so A : 0
Nhap so B : 0
A / B = NaN
-- program is finished running --
```



Tổng số lệnh: 34

Trong đó:

- R-type: 7 lệnh chiếm 20%.
- I-type: 25 lệnh chiếm 73%.
- J-type: 2 lệnh chiếm 6%.

Tính thời gian chạy:

$$\text{CPU time} = \frac{34 \times 1}{1.8 \times 10^9} = 0.0189 \times 10^{-6} \text{ (s)} = 0.0189 \text{ (}\mu\text{s)}$$

## Tài liệu tham khảo

- [1] *Phạm Quốc Cường, Kiến trúc máy tính, Nhà xuất bản đại học quốc gia thành phố Hồ Chí Minh, Năm 2017.*
- [2] *MIPS Technologies, Inc MIPS32™ Architecture For Programmers Volume II: The MIPS32™ Instruction Set. 2003.*
- [3] *Hợp ngữ MIPS: <https://vietcodes.github.io/algo/mips>.*
- [4] *Float Point Tutorial:  
[https://www.rfwireless-world.com/Tutorials/floating-point-tutorial.html?](https://www.rfwireless-world.com/Tutorials/floating-point-tutorial.html?fbclid)  
[fbclid](https://www.rfwireless-world.com/Tutorials/floating-point-tutorial.html?fbclid).*