

ĐẠI HỌC BÁCH KHOA THÀNH PHỐ HỒ CHÍ MINH
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



KIẾN TRÚC MÁY TÍNH

Nhóm: L01_05 — Bài tập lớn

Kiến trúc tập lệnh MIPS

Giáo viên hướng dẫn: Nguyễn Xuân Minh

Sinh viên: Nguyễn Đăng Hải
Lê Thanh Minh

1913254
Không tham gia

Thành phố Hồ Chí Minh, 12/2020

Mục lục

1	Đề bài	2
2	Yêu cầu	2
3	Ý tưởng hiện thực	2
4	Trình tự hiện thực	2
5	Thông kê số lệnh, loại lệnh (instruction type) và thời gian chạy	3
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	7
5.6	Ví dụ 6	7
5.7	Ví dụ 7	8
	Tài liệu tham khảo	9

1 Đề bài

Chia 2 số thực

Cho 2 số thực dạng chuẩn (Standard Floating Point) A và B với độ chính xác đơn (32 bit). Sử dụng hợp ngữ assembly MIPS, viết thủ tục chia hai số A, B. A và B được hiểu là số chia (dividend) và số bị chia (divisor).

2 Yêu cầu

- Sử dụng tập lệnh MIPS để hiện thực các thủ tục bên dưới.
- Thông kê số lệnh, loại lệnh (instruction type) của mỗi chương trình.
- Tính và trình bày cách tính thời gian chạy của chương trình trên máy tính kiến trúc MIPS. Máy đang sử dụng có tần số 1.6 GHz up to 3.4 GHz (i5 - 8250U), cho nên sẽ lấy chuẩn là 3.4 GHz để tính toán trong báo cáo này.
- Code:
 - Code style phải rõ ràng, có comment, phân hoạch công việc theo từng hàm, **CHỈ DÙNG LỆNH MIPS CHUẨN**.
 - Truyền nhận và trả kết quả khi gọi hàm theo quy ước của thanh ghi (thanh ghi \$a argument, thanh ghi \$v giá trị trả về khi gọi hàm).
 - Xuất kết quả để kiểm tra.
- Báo cáo: Nộp 2 files:
 - Diễn giải (file .PDF)
 - Chương trình (file .ASM)

3 Ý tưởng hiện thực

Để chia hai số thực mà chỉ dùng lệnh MIPS chuẩn thì ta sẽ đưa hai số thực cần tính toán về định dạng IEEE 754. 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 hiện thực

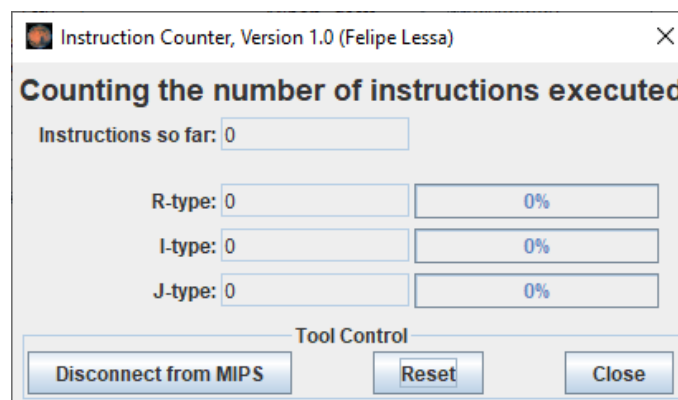
Chia 2 số thực bất kỳ ta thực hiện theo trình tự sau:

- Nhập và lưu giá trị của 2 số thực *dividend* và *divisor*. Sau đó load giá trị vào các thanh ghi
- Kiểm tra 3 trường hợp đặc biệt:
 - dividend* = 0 và *divisor* = 0. Khi đó, ta không thể thực hiện phép chia. Nên $dividend/divisor = NaN$
 - dividend* = 0 và *divisor* \neq 0. Đây là phép chia của 0 cho một số thực bất kỳ vậy kết quả: $dividend/divisor = 0$

- $dividend \neq 0$ và $divisor = 0$. Đây là phép chia của một số thực khác 0 cho 0. Vậy kết quả sẽ là $dividend/divisor = \text{infinity}$
- Nếu không phải 1 trong 3 trường hợp trên tức là $dividend \neq 0$ và $divisor \neq 0$ thì ta sẽ đưa cả 2 số thực về dạng IEEE 754 như sau:
 - Dịch dividend và divisor 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 dividend và divisor 1 bit để bỏ bit dấu. Sau đó dịch phải 24 bit. Ta sẽ thu được 8 bit của phần mũ E.
 - Dịch trái dividend và divisor 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 dividend và divisor dùng phép XOR để xác định dấu của phép chia
 - Xác định phần mũ bằng cách. Lấy phần mũ của dividend trừ đi divisor cộng bias (127). Tức là: $E_{dividend} - E_{divisor} + bias$
 - 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 0x007FFFFF 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. Theo yêu cầu là $3.4GHz$ hay trong một giây có 3.4×10^9 dao động.

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

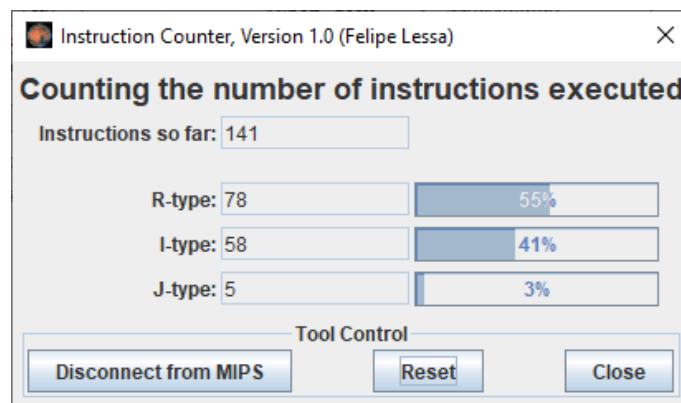
5.1 Ví dụ 1

Nhập vào: $dividend = 6.75$ và $divisor = 2.0$ kết quả của phép chia là $Thuong : 3.375$ Tổng số lệnh: 141

Trong đó:

- R-type: 78 lệnh chiếm 55%
- I-type: 58 lệnh chiếm 41%
- J-type: 5 lệnh chiếm 3%

```
Nhap so bi chia: 6.75
Nhap so chia: 2.0
Thuong: 3.375
-- program is finished running --
```



Tính toán thời gian chạy:

$$CPU \text{ time} = \frac{\text{Instruction Count} \times CPI}{\text{Clock rate}} = \frac{141 \times 1}{3.4 \times 10^9} = 0.0415 \times 10^{-6} s = 0.415 \mu s$$

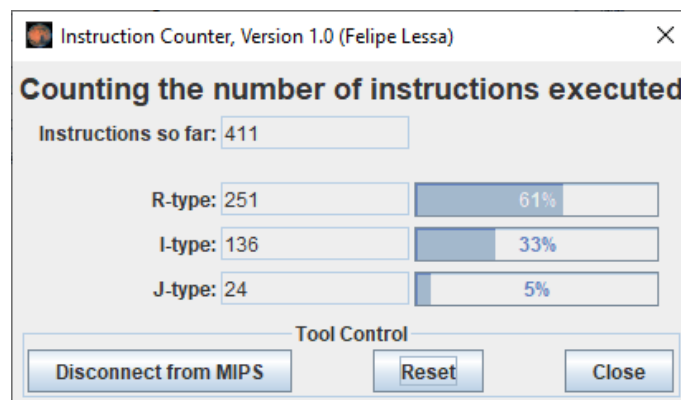
5.2 Ví dụ 2

Nhập vào: $dividend = 23.1$ và $divisor = 3.45$ kết quả của phép chia là $Thuong : 6.695652$ Tổng số lệnh: 411

Trong đó:

- R-type: 251 lệnh chiếm 61%
- I-type: 136 lệnh chiếm 33%
- J-type: 24 lệnh chiếm 5%

```
Nhap so bi chia: 23.1
Nhap so chia: 3.45
Thuong: 6.695652
-- program is finished running --
```



Tính toán thời gian chạy:

$$\text{CPU time} = \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock rate}} = \frac{411 \times 1}{3.4 \times 10^9} = 0.1209 \times 10^{-6} \text{ s} = 0.1209 \mu\text{s}$$

5.3 Ví dụ 3

Nhập vào: $dividend = 0$ và $divisor = 43.2$ kết quả của phép chia là $Thuong : 0$ Tổng số lệnh: 32

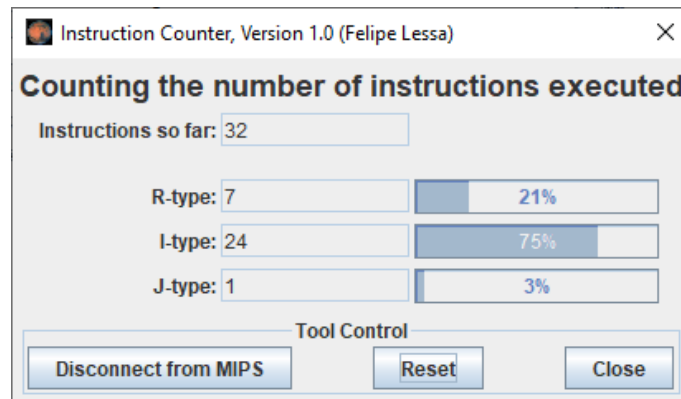
Trong đó:

- R-type: 7 lệnh chiếm 21%
- I-type: 24 lệnh chiếm 75%
- J-type: 1 lệnh chiếm 3%

Tính toán thời gian chạy:

$$\text{CPU time} = \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock rate}} = \frac{32 \times 1}{3.4 \times 10^9} = 0.0094 \times 10^{-6} \text{ s} = 0.0094 \mu\text{s}$$

```
Nhap so bi chia: 0
Nhap so chia: 43.2
Thuong: 0
-- program is finished running --
```



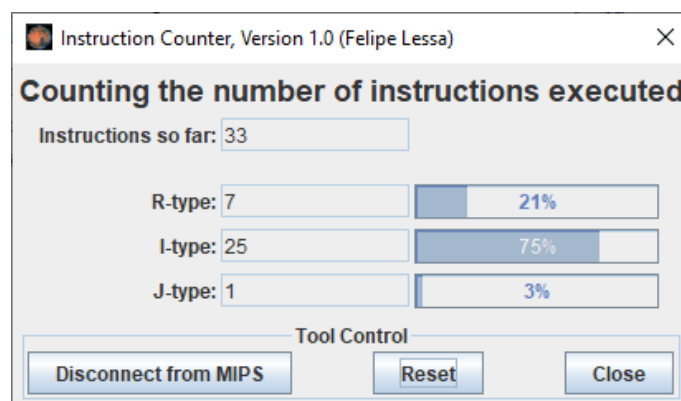
5.4 Ví dụ 4

Nhập vào: $dividend = -14.2$ và $divisor = 0$ kết quả của phép chia là $Thuong : Infinity$ Tổng số lệnh: 33

Trong đó:

- R-type: 7 lệnh chiếm 21%
- I-type: 25 lệnh chiếm 75%
- J-type: 1 lệnh chiếm 3%

```
Nhap so bi chia: -14.2
Nhap so chia: 0
Thuong: Infinity
-- program is finished running --
```



Tính toán thời gian chạy:

$$CPU \text{ time} = \frac{\text{Instruction Count} \times CPI}{\text{Clock rate}} = \frac{33 \times 1}{3.4 \times 10^9} = 0.0097 \times 10^{-6} s = 0.0097 \mu s$$

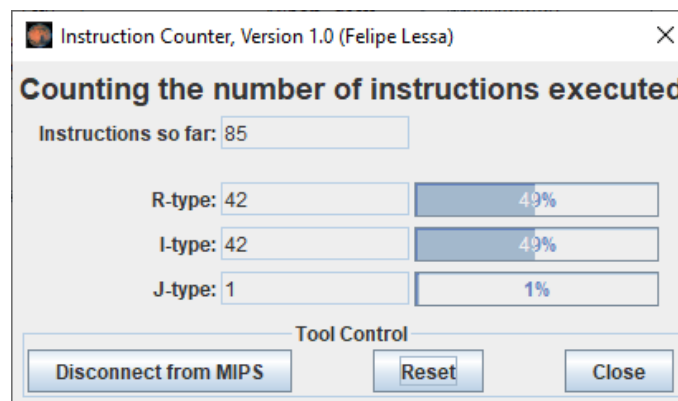
5.5 Ví dụ 5

Nhập vào: $dividend = -15$ và $divisor = 7.5$ kết quả của phép chia là $Thuong : -2.0$ Tổng số lệnh: 85

Trong đó:

- R-type: 42 lệnh chiếm 49%
- I-type: 42 lệnh chiếm 49%
- J-type: 1 lệnh chiếm 1%

```
Nhap so bi chia: -15
Nhap so chia: 7.5
Thuong: -2.0
-- program is finished running --
```



Tính toán thời gian chạy:

$$CPU \text{ time} = \frac{\text{Instruction Count} \times CPI}{\text{Clock rate}} = \frac{85 \times 1}{3.4 \times 10^9} = 0.025 \times 10^{-6}s = 0.025\mu s$$

5.6 Ví dụ 6

Nhập vào: $dividend = 3.3$ và $divisor = -9.9$ kết quả của phép chia là $Thuong : -0.33333334$ Tổng số lệnh: 409

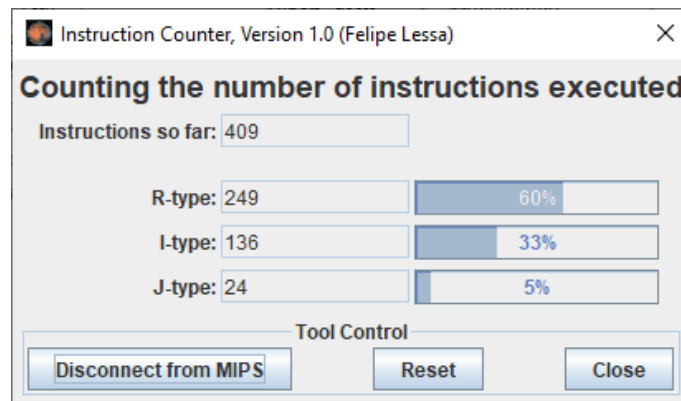
Trong đó:

- R-type: 249 lệnh chiếm 60%
- I-type: 136 lệnh chiếm 33%
- J-type: 24 lệnh chiếm 5%

```
Nhap so bi chia: 3.3
Nhap so chia: -9.9
Thuong: -0.33333334
-- program is finished running --
```

Tính toán thời gian chạy:

$$CPU \text{ time} = \frac{\text{Instruction Count} \times CPI}{\text{Clock rate}} = \frac{409 \times 1}{3.4 \times 10^9} = 0.1203 \times 10^{-6}s = 0.1203\mu s$$

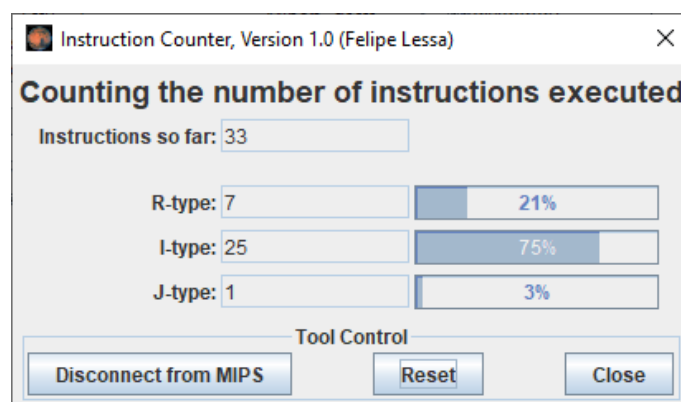


5.7 Ví dụ 7

Nhập vào: $dividend = 0$ và $divisor = 0$ kết quả của phép chia là *Thuong : NaN* Tổng số lệnh: 33
Trong đó:

- R-type: 7 lệnh chiếm 21%
- I-type: 25 lệnh chiếm 75%
- J-type: 1 lệnh chiếm 3%

```
Nhap so bi chia: 0
Nhap so chia: 0
Thuong: NaN
-- program is finished running --
```



Tính toán thời gian chạy:

$$\text{CPU time} = \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock rate}} = \frac{33 \times 1}{3.4 \times 10^9} = 0.00971 \times 10^{-6} \text{s} = 0.00971 \mu\text{s}$$



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 2019.
- [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?fbclid>.