

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



KIẾN TRÚC MÁY TÍNH (CO2008)

---

BÀI TẬP LỚN

Đề 2

Cộng 2 số thực chuẩn  
IEEE 754 chính xác đơn.

---

Nhóm: **65**  
*Đỗ Chí Khải* - 1913769  
*Trương Thành Long* - 2211909  
GV: *Nguyễn Xuân Minh*  
Lớp: L02



## Danh sách thành viên:

STT	Tên	MSSV
1	Đỗ Chí Khải	1913769
2	Trương Thành Long	2211909

## Contents

<b>1</b>	<b>Đề bài</b>	<b>3</b>
<b>2</b>	<b>Giải pháp hiện thực</b>	<b>3</b>
2.1	Cơ sở lý thuyết . . . . .	3
2.1.1	Số thực chuẩn IEEE 754 . . . . .	3
2.1.2	Vô cùng, zeros, số rất nhỏ . . . . .	4
	Zero . . . . .	4
	Subnormals . . . . .	4
	Vô cùng . . . . .	4
2.1.3	Làm tròn (round to nearest, ties to even) . . . . .	4
2.2	Các bước thực hiện . . . . .	5
<b>3</b>	<b>Giải thuật</b>	<b>6</b>
3.1	Sơ đồ hiện thực . . . . .	6
3.2	Sơ đồ bước làm tròn bằng bit RS . . . . .	7
<b>4</b>	<b>Thông kê</b>	<b>8</b>
4.1	Cộng 2 số thực cùng dấu (3.123334 và 2.324324) . . . . .	8
4.1.1	Kết quả kiểm thử . . . . .	8
4.1.2	Tính thời gian chạy chương trình . . . . .	8
4.2	Cộng 2 số thực khác dấu (-3.4454353 & 2.3212311) . . . . .	9
4.2.1	Kết quả kiểm thử . . . . .	9
4.2.2	Tính thời gian chạy chương trình . . . . .	9
4.3	Cộng 2 số rất lớn (Infinity) (3.3E38 & 1.3E38) . . . . .	10
4.3.1	Kết quả kiểm thử . . . . .	10
4.3.2	Tính thời gian chạy chương trình . . . . .	10
4.4	Cộng 2 số rất nhỏ (Zeros) (1.4E-45 & 1.4E-45)) . . . . .	11
4.4.1	Kết quả kiểm thử . . . . .	11
4.4.2	Tính thời gian chạy chương trình . . . . .	11
4.5	Cộng 2 số có độ chênh lệch lớn (2.5 & 448.00043) . . . . .	12
4.5.1	Kết quả kiểm thử . . . . .	12
4.5.2	Tính thời gian chạy chương trình . . . . .	12
<b>5</b>	<b>Tài liệu tham khảo</b>	<b>13</b>

## 1 Đề bài

**Đề 2:** Viết chương trình thực hiện phép cộng 2 số thực chuẩn IEEE 754 chính xác đơn mà không dùng các lệnh tính toán số thực của MIPS. Dữ liệu đầu vào đọc từ file lưu trữ dạng nhị phân trên đĩa FLOAT2.BIN (2 trị  $\times$  4 bytes = 8 bytes).

## 2 Giải pháp hiện thực

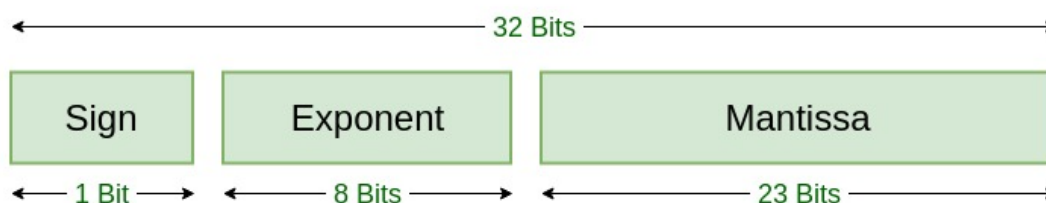
### 2.1 Cơ sở lý thuyết

#### 2.1.1 Số thực chuẩn IEEE 754

Chuẩn IEEE cho số dấu phẩy động (IEEE 754) là một tiêu chuẩn kỹ thuật tính toán dấu phẩy động được thực hiện vào năm 1985 bởi Viện kỹ sư điện và điện tử (IEEE). Chuẩn IEEE 754 là cách biểu diễn phổ biến nhất hiện nay cho các số thực trên máy tính, bao gồm PC trên nền Intel, máy Mac và hầu hết các nền tảng Unix.

Có một số cách để biểu diễn số dấu phẩy động nhưng IEEE 754 là hiệu quả nhất trong hầu hết các trường hợp. IEEE 754 có 3 thành phần cơ bản:

- **Dấu:** 0 đại diện cho số dương trong khi 1 đại diện cho số âm.
- **Số mũ:** Trường số mũ cần đại diện cho cả số mũ dương và âm. Độ lệch được thêm vào số mũ để có được số mũ để lưu trữ.
- **Định trị:** Phần định trị là một phần của một số trong ký hiệu khoa học hoặc số dấu phẩy động, bao gồm các chữ số có nghĩa của nó. Ở đây, chúng tôi chỉ có 2 chữ số, tức là 0 và 1. Vì vậy, một phần định trị được chuẩn hóa là một phần chỉ có một số 1 ở bên trái của số thập phân.



### Single Precision IEEE 754 Floating-Point Standard

Hình 1: Số thực chuẩn IEEE 754

### 2.1.2 Vô cùng, zeros, số rất nhỏ

#### Zero

Số 0 không thể được chuẩn hóa, cho nên chúng ta không thể biểu diễn các số thực ấy dưới dạng chuẩn hóa IEEE 754. Nếu 1 số có toàn bộ các bit đều là 0 thì số đó biểu diễn cho số 0. Chúng ta cũng có thể biểu số 0 cùng với bit dấu, nhưng  $-0$  và  $0$  đều được xem như nhau.

#### Subnormals

Chúng ta cũng có thể hiển thị các giá trị nhỏ hơn giá trị nhỏ nhất có thể hiển thị. Các số đó được gọi là số rất nhỏ (Subnormals). Nếu phần mũ đều là bit 0 và định trị khác 0. Trong bài báo cáo này, để dễ dàng thì các số rất nhỏ sẽ được coi như là số 0.0.

#### Vô cùng

Giá trị  $+\infty$  và  $-\infty$  được biểu diễn bởi các bit mũ đều là 1, các bit định trị đều là 0. m hay dương vô cùng sẽ được xác định bằng bit dấu.

### 2.1.3 Làm tròn (round to nearest, ties to even)

#### Rounding



#### • Round up conditions

- Round = 1, Sticky = 1  $\Rightarrow > 0.5$
- Guard = 1, Round = 1, Sticky = 0  $\Rightarrow$  Round to even

Value	Fraction	GRS	Incr?	Rounded
128	1.0000000	000	N	1.000
15	1.1010000	100	N	1.101
17	1.0001000	010	N	1.000
19	1.0011000	110	Y	1.010
138	1.0001010	011	Y	1.001
63	1.1111100	111	Y	10.000

Systems Programming 2022 Ch. 13: Floating Point



Khi chúng ta thực hiện việc cắt 1 số lượng bit ở phía bên phải của số thực dạng IEEE 754, ta cần phải chú ý đến việc có cần làm tròn nó không. Có nhiều cách làm tròn khác nhau, làm tròn lên, xuống, hoặc chính xác nhất là làm tròn đến số gần nhất, còn nếu số đó nằm ngay giữa thì làm tròn đến số bit chẵn.

G (Guard bit) là bit thấp nhất của các bit sau khi bị cắt, R (Round) là bit tiếp theo bên phải và cũng là bit đầu tiên của các bit bị cắt, cuối cùng là S (Sticky) bit là bit bên phải bit R và là kết quả của phép OR các số còn lại bên phải bit R. 3 bit này sẽ được sử dụng để thực hiện việc làm tròn.

Để làm tròn lên, chúng ta cộng 1 vào bit thấp nhất bên trái bit G.

## 2.2 Các bước thực hiện

*Step 0:* Mở file và đọc các giá trị dữ liệu.

*Step 1:* Trích xuất dấu, phần mũ, định trị. Thêm bit 1 dẫn vào đầu định trị.

*Step 2:* Kiểm tra xem mũ có bằng nhau không, nếu không thì dịch định trị, điều chỉnh mũ của số có mũ nhỏ cho đến khi mũ bằng với mũ của số có mũ lớn.

*Step 3:* Nếu các số hạng cùng dấu cộng 2 định trị, dấu kết quả dựa vào dấu 2 định trị. Nếu khác dấu thì đổi dấu số âm rồi cộng 2 định trị, nếu kết quả của phép cộng đó là âm thì tiếp tục đổi dấu.

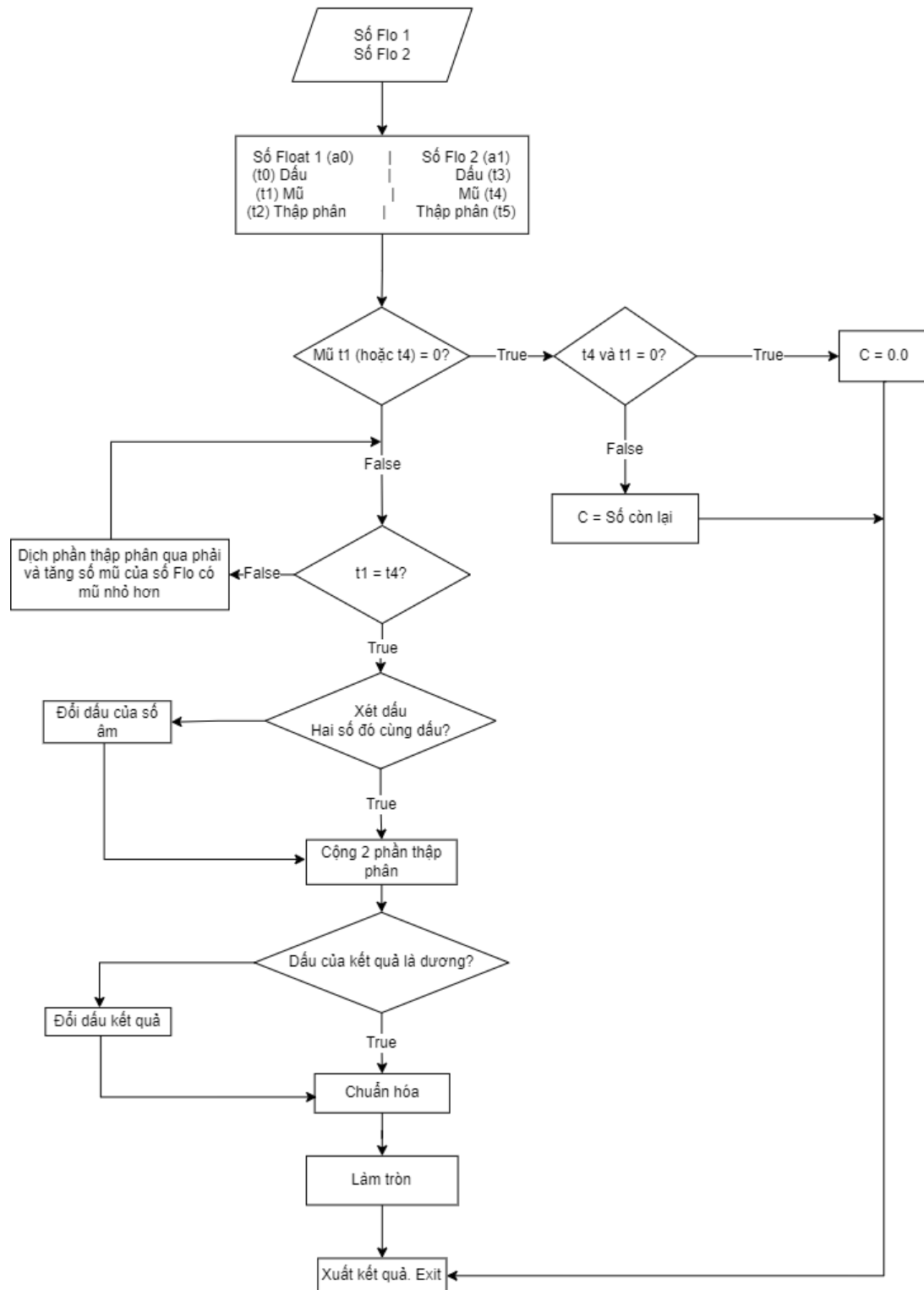
*Step 4:* Tiến hành chuẩn hóa kết quả, nếu sau khi cộng định trị mà có bit carry thì tăng mũ, dịch phải định trị. Nếu định trị không có bit 1 ở đầu thì dịch trái định trị, giảm số mũ cho đến khi có được bit 1 ở đầu.

*Step 5:* Làm tròn kết quả theo quy định của chuẩn IEEE 754. Sau khi làm tròn, nếu có bit carry thì dịch phải, tăng mũ, nếu mũ = 1111 1111 thì kết quả là vô cùng

*Step 6:* Cắt bỏ bit trái cùng của định trị, ghép các thành phần và xuất kết quả

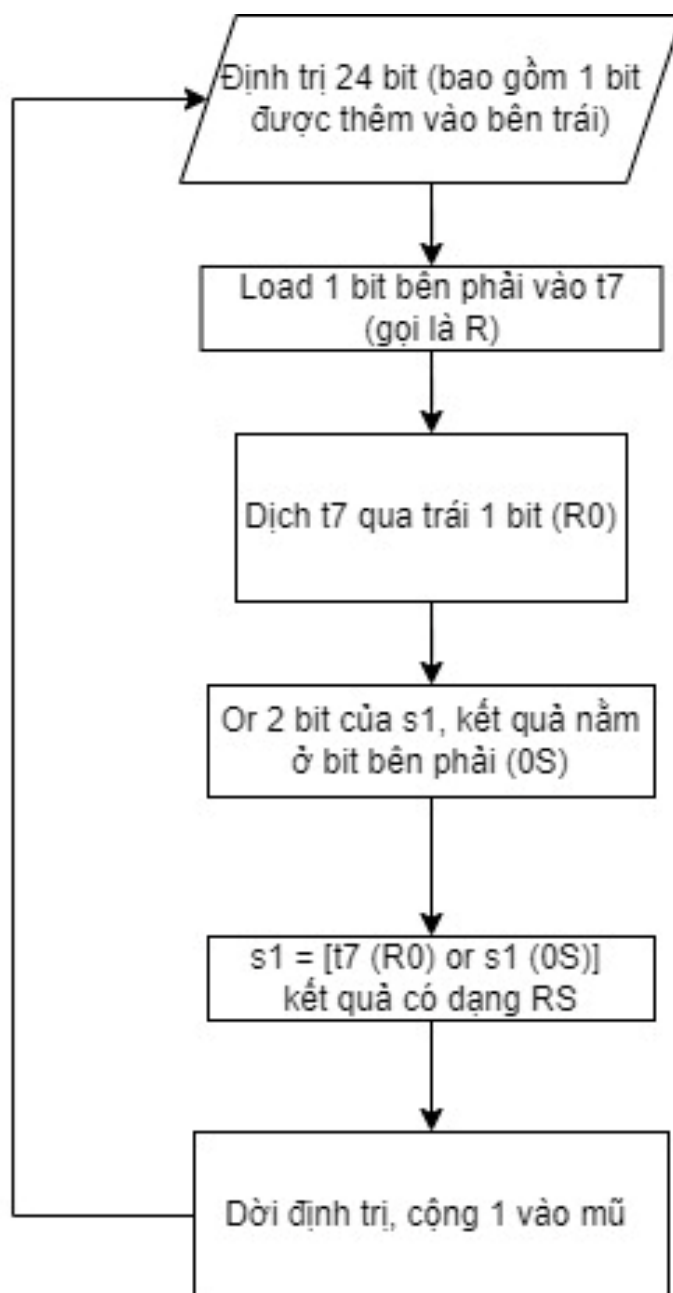
## 3 Giải thuật

### 3.1 Sơ đồ hiện thực



Hình 2: Sơ đồ hiện thực

### 3.2 Sơ đồ bước làm tròn bằng bit RS



Hình 3: Sơ đồ bước làm tròn bằng bit RS



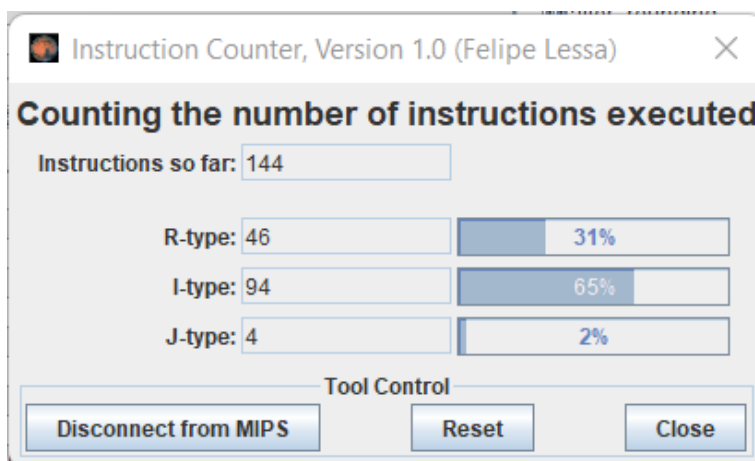
## 4 Thống kê

### 4.1 Cộng 2 số thực cùng dấu (3.123334 và 2.324324)

#### 4.1.1 Kết quả kiểm thử

```
Du lieu 1 = 3.123334
Du lieu 2 = 2.324324
Ket qua   = 5.4476576
-- program is finished running --
```

#### 4.1.2 Tính thời gian chạy chương trình



Với mô hình đơn chu kỳ Single Cycle,  $CPI = 1$ , Clock Rate = 1GHz, thời gian thực thi của chương trình được tính theo công thức:

$$CPUTime = \frac{InstructionsCount \times CPI}{ClockRate} = \frac{144 \times 1}{10^9} = 144(ns)$$

## 4.2 Cộng 2 số thực khác dấu (-3.4454353 & 2.3212311)

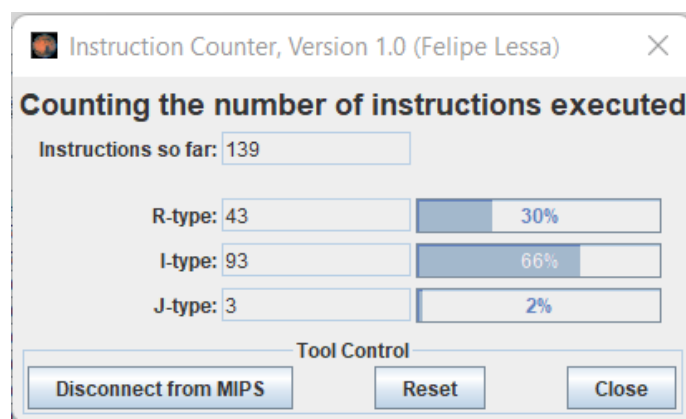
### 4.2.1 Kết quả kiểm thử

```

Du lieu 1 = -3.4454353
Du lieu 2 = 2.3212311
Ket qua   = -1.1242042
-- program is finished running --

```

### 4.2.2 Tính thời gian chạy chương trình



Với mô hình đơn chu kỳ Single Cycle,  $CPI = 1$ , Clock Rate = 1GHz, thời gian thực thi của chương trình được tính theo công thức:

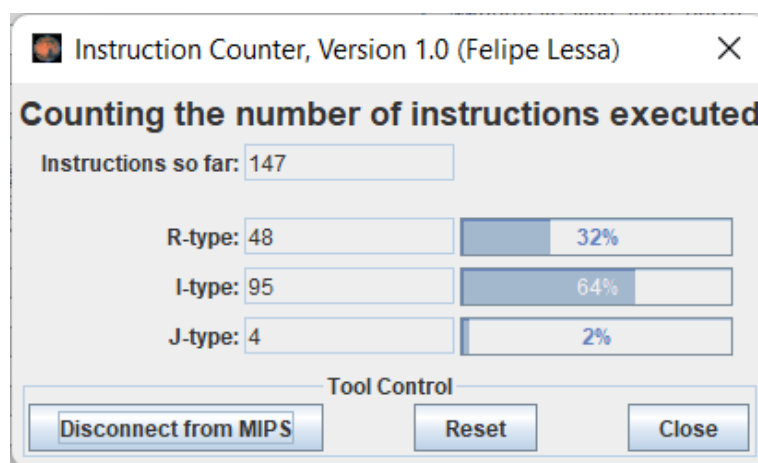
$$CPUTime = \frac{InstructionsCount \times CPI}{ClockRate} = \frac{139 \times 1}{10^9} = 139(ns)$$

### 4.3 Cộng 2 số rất lớn (Infinity) (3.3E38 & 1.3E38)

#### 4.3.1 Kết quả kiểm thử

```
Du lieu 1 = 3.3E38
Du lieu 2 = 1.3E38
Ket qua    = Infinity
-- program is finished running --
```

#### 4.3.2 Tính thời gian chạy chương trình



Với mô hình đơn chu kỳ Single Cycle,  $CPI = 1$ , Clock Rate = 1GHz, thời gian thực thi của chương trình được tính theo công thức:

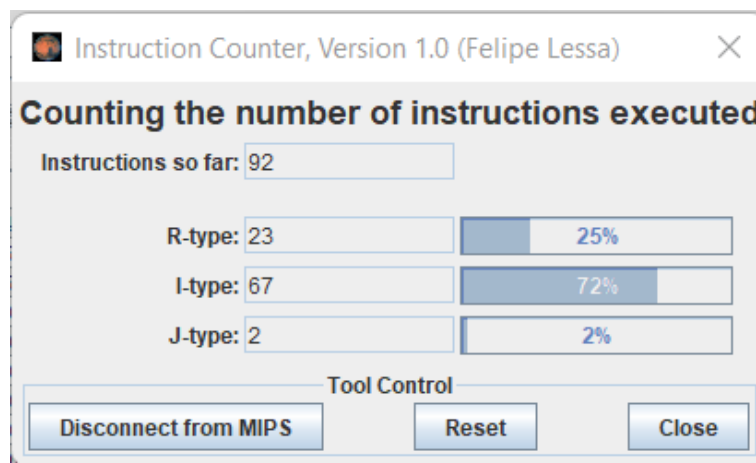
$$CPUTime = \frac{InstructionsCount \times CPI}{ClockRate} = \frac{147 \times 1}{10^9} = 147(ns)$$

## 4.4 Cộng 2 số rất nhỏ (Zeros) (1.4E-45 & 1.4E-45))

### 4.4.1 Kết quả kiểm thử

```
Du lieu 1 = 1.4E-45
Du lieu 2 = 1.4E-45
Ket qua    = 0.0
-- program is finished running --
```

### 4.4.2 Tính thời gian chạy chương trình



Với mô hình đơn chu kỳ Single Cycle,  $CPI = 1$ , Clock Rate = 1GHz, thời gian thực thi của chương trình được tính theo công thức:

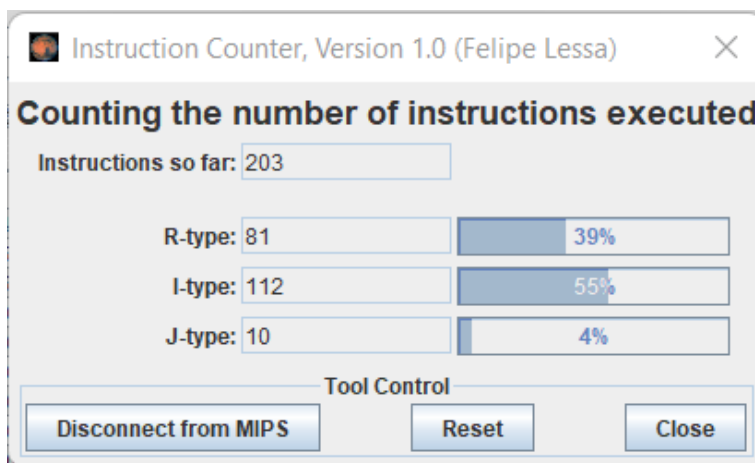
$$CPUTime = \frac{InstructionsCount \times CPI}{ClockRate} = \frac{92 \times 1}{10^9} = 92(ns)$$

## 4.5 Cộng 2 số có độ chênh lệch lớn (2.5 & 448.00043)

### 4.5.1 Kết quả kiểm thử

```
Du lieu 1 = 2.5
Du lieu 2 = 448.00043
Ket qua   = 450.50043
-- program is finished running --
```

### 4.5.2 Tính thời gian chạy chương trình



Với mô hình đơn chu kỳ Single Cycle,  $CPI = 1$ , Clock Rate = 1GHz, thời gian thực thi của chương trình được tính theo công thức:

$$CPUTime = \frac{InstructionsCount \times CPI}{ClockRate} = \frac{203 \times 1}{10^9} = 203(ns)$$

## 5 Tài liệu tham khảo

- [1] Các slides và tài liệu thầy up trên BK-el
- [2] Advanced MIPS coding, *utdallas.edu*, link.
- [3] X07-Floating-Point.ppt, *live.com*, link.
- [4] Understanding Overflow through Examples, *vlsifacts.com*, link
- [5] IEEE Floating-Point Representation, *learn.microsoft.com*, link
- [6] Floating Point Systems, *sci.utah.edu*, link
- [7] Understanding Overflow through Examples, *vlsifacts.com*, link
- [8] IEEE 754 Calculator