

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



BÁO CÁO BÀI TẬP LỚN KIẾN TRÚC MÁY TÍNH
Kiến trúc tập lệnh
Đề 2

Giảng viên hướng dẫn: Nguyễn Xuân Minh

Sinh viên: Cao Vĩnh Phát (2212794)
Nguyễn Quang Nghiêm (2212209)

Ho Chi Minh, Ngày 19 tháng 5 năm 2024

Mục lục

1	Giới thiệu đề tài	2
2	Giải thuật	3
2.1	Sơ lược hướng đi	3
2.2	Chi tiết giải thuật	3
2.2.1	Lấy các thông tin fraction, exponent và S từ thanh ghi	3
2.2.2	Cộng và trừ hai thanh ghi fraction	3
2.2.3	Lấy từng thành phần của kết quả	4
2.2.4	Cộng các số đã tính toán	4
3	Đánh giá chương trình	5
4	Hình ảnh INPUT/OUTPUT của các testcase	6

1 Giới thiệu đề tài

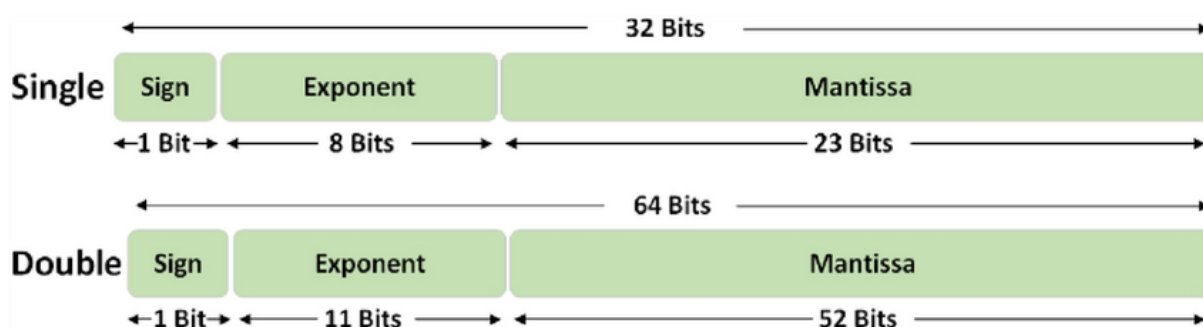
Đề 2:

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 x 4 bytes = 8 bytes).

Số thực dạng chuẩn IEEE 754 là gì?

Cần biết rằng, máy tính chỉ có thể lưu trữ được các tín hiệu nhị phân, do đó, để biểu diễn một số thực vào trong máy tính, chúng ta phải chuyển nó thành 1 chuỗi nhị phân. Tuy nhiên không thể biểu diễn số thực theo cách thông thường giống như số nguyên mà cần có một chuẩn riêng, phổ biến nhất là chuẩn IEEE 754.

Số thực được biểu diễn dưới dạng IEEE 754 có độ chính xác đơn (được biểu diễn bằng 32 bit (1 bit dấu – 8 bit mũ – 23 bit giá trị phần lẻ) và độ chính xác kép (được biểu diễn bằng 64 bit (1 bit dấu – 11 bit mũ – 52 bit giá trị phần lẻ).



Hình 1: *IEEE Standard for Floating-Point Arithmetic (IEEE 754)*

Để có thể lưu được số thực dưới dạng IEEE 754, chúng ta cần phải chuyển số chúng ta cần lưu về dạng số thực chuẩn (số thực chuẩn là số mà phần nguyên không có chứa số 0).

2 Giải thuật

2.1 Sơ lược hướng đi

Để cộng (trừ) hai số thực dưới định dạng IEEE 754, ta cần đưa hai số thực X và Y từ thanh ghi về dạng chuẩn.

$$X = 1.xxx...x \times 2^{(exponentX-127)} \quad (1)$$

$$Y = 1.xxx...x \times 2^{(exponentY-127)} \quad (2)$$

Chú thích: x có thể là 0 hoặc 1 và có tổng cộng 23 bit (tương ứng với 23 bit phần thập phân trong IEEE 754).

Bước tiếp theo ta cần dịch phải số có exponent nhỏ hơn sao cho X và Y có phần $2^{exponent-127}$ bằng nhau.

Đặt $2^{exponent-127}$ làm thừa số chung, cộng trừ kết quả và chuẩn hoá về dạng IEEE 754 (nếu cần).

2.2 Chi tiết giải thuật

2.2.1 Lấy các thông tin fraction, exponent và S từ thanh ghi

1. Extract Sign (trích xuất bit dấu)

Hàm `extractSign` được sử dụng để trích xuất bit dấu từ một số dấu chấm động. Cụ thể, hàm này sẽ dịch phải 31 bit để trích xuất bit dấu và lưu trữ kết quả vào thanh ghi `$v0`.

2. Extract Biased Exponent (trích xuất bit exponent)

Hàm `extractBiasedExponent` trong MIPS trích xuất phần mũ từ một số dấu chấm động. Quá trình này bao gồm việc dịch trái một bit, sau đó dịch phải 24 bit từ số dấu chấm động. Kết quả được lưu vào thanh ghi `$v0`.

3. Extract Fraction (trích xuất bit fraction)

Hàm `extractFraction` trong MIPS được sử dụng để trích xuất phần fraction từ một số dấu chấm động. Quá trình này bao gồm việc dịch trái 9 bit, sau đó dịch phải 9 bit từ số dấu chấm động và cuối cùng là cộng với `0x800000` để thêm `1b'1` và ta được dạng $(1 + \text{fraction})$. Kết quả được lưu vào thanh ghi `$v0`.

2.2.2 Cộng và trừ hai thanh ghi fraction

Sau khi đã lấy được các thông tin cơ bản của hai số thực, ta tiến hành phép cộng(trừ) như sau:

Trường hợp 1:

Khi 2 giá trị có cùng dấu thì ta tiến hành cộng 2 thanh ghi fraction mà ta vừa trích xuất được.

Sau khi có được kết quả thì ta xử lý trường hợp overflow bằng cách dịch phải 24 bits

để so sánh với giá trị 0. Nếu giá trị được so sánh bằng 0 (underflow) thì ta không làm gì cả, ngược lại nếu khác 0 (overflow) thì ta tiến hành xử lý bằng cách dịch phải 1 bit.

Trường hợp 2:

Khi 2 giá trị có trái dấu thì ta tiến hành lấy thanh ghi fraction có giá trị lớn hơn trừ thanh ghi fraction có giá trị nhỏ hơn và dấu của kết quả sẽ mang dấu của giá trị lớn hơn.

Sau đó ta tiến hành chuẩn hoá kết quả vừa có được để trở thành dạng chuẩn IEEE 754 (nếu cần).

2.2.3 Lấy từng thành phần của kết quả

Sau khi cộng 2 thanh ghi fraction và có được kết quả thì ta tiến hành dịch trái 9 bit và dịch phải 9 bit của thanh ghi fraction vừa có để có được 1 thanh ghi chứa fraction (bit 10 - bit 32).

Ta có 1 thanh ghi chứa dấu, ta tiến hành dịch phải 31 bit để có được 1 thanh ghi chứa dấu (bit 1).

Ta có thanh ghi chứa BiasedExponent, ta tiến hành dịch trái 23 bit để có được thanh ghi chứa BiasedExponent (bit 2 - bit 9).

2.2.4 Cộng các số đã tính toán

Sau khi có từng thành phần thì ta tiến hành cộng 3 thanh ghi vừa có được.

Sau đó ta chuẩn hoá kết quả vừa có được (nếu cần). Sau đó lưu kết quả vào thanh ghi \$v0.

3 Đánh giá chương trình

a	b	ourcode	MIPS	R	I	J	IC	Time(ns)
162.2	32.2	194.4	194.4	82	78	15	175	175
535.564	523.456	1509.02	1509.02	77	73	13	163	163
34.5	754.8	789.3	789.3	83	81	14	178	178
712.567	42.675	755.242	755.242	83	81	13	177	177
25346.123	3457.346	28803.469	28803.469	82	79	13	174	174
-139.46	-123.42	-262.88	-262.88	82	77	15	174	174
-23.5	-49.5	-73.0	-73.0	82	77	16	175	175
-48.7	-72.5	-121.2	-121.2	81	76	16	173	173
-123.5	-49	-172.5	-172.5	146	154	30	348	348
-129.4	-37.6	-166.99998	-167.0	82	78	15	175	175
-34.644	345.63	310.98602	310.986	82	82	14	178	178
456	-24	432.0	432.0	80	81	14	183	183
234.64	-53.75	180.89	180.89	81	80	12	173	173
-346.324	345.664	-0.66000366	-0.66000366	103	111	21	235	235
1568.23	-234.6	1333.63	1333.63	82	82	12	176	176

4 Hình ảnh INPUT/OUPUT của các testcase

testcase 1:

```
162.2
32.2
result from our code: 194.4
result from mips float instruction: 194.4
```

testcase 2:

```
535.564
523.456
result from our code: 1059.02
result from mips float instruction: 1059.02
```

testcase 3:

```
34.5
754.8
result from our code: 789.3
result from mips float instruction: 789.3
```

testcase 4:

```
712.567
42.675
result from our code: 755.242
result from mips float instruction: 755.242
```

testcase 5:

```
25346.123
3457.346
result from our code: 28803.469
result from mips float instruction: 28803.469
```

testcase 6:

```
**** user input : -139.46
**** user input : -123.42
result from our code: -262.88
result from mips float instruction: -262.88
```

testcase 7:

```
**** user input : -23.5
**** user input : -49.5
result from our code: -73.0
result from mips float instruction: -73.0
```

testcase 8:

```
**** user input : -48.7
**** user input : -72.5
result from our code: -121.2
result from mips float instruction: -121.2
```

testcase 9:

```
**** user input : -123.5
**** user input : -49
result from our code: -172.5
result from mips float instruction: -172.5
```

testcase 10:

```
**** user input : -129.4
**** user input : -37.6
result from our code: -166.99998
result from mips float instruction: -167.0
```




testcase 11:

```
-34.644  
345.63  
result from our code: 310.98602  
result from mips float instruction: 310.986
```

testcase 12:

```
456  
-24  
result from our code: 432.0  
result from mips float instruction: 432.0
```

testcase 13:

```
234.64  
-53.75  
result from our code: 180.89  
result from mips float instruction: 180.89
```

testcase 14:

```
-346.324  
345.664  
result from our code: -0.66000366  
result from mips float instruction: -0.66000366
```

testcase 15:

```
1568.23  
-234.6  
result from our code: 1333.63  
result from mips float instruction: 1333.63
```