

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



CHIA HAI SỐ THỰC DẦU CHẤM ĐỘNG TRONG HỢP NGỮ MIPS

Thành viên nhóm	Mã số sinh viên	Lớp
Lê Minh Nghĩa	2010445	L10
Võ Thái Toàn	2010709	L05
Bùi Khánh Vĩnh	2010091	L02
Trần Nguyên Vũ	2012445	L05

Thành phố Hồ Chí Minh, Tháng 11 năm 2020

1 Cơ sở lý thuyết

1.1 Biểu diễn số thực

Trong thực tiễn, phần lớn các phép tính toán đều có sự xuất hiện của số thực. Số thực có thể được biểu diễn bằng nhiều cách. Mà trong đó, chúng ta quan tâm đến cách biểu diễn số thực theo ký hiệu khoa gọi (*scientific notation*). Trong cách biểu diễn này, số thực sẽ được chia ra làm hai phần, đó là phần hệ số (*coefficient*) và phần mũ (*exponent*). Trong khi biểu diễn một số thực theo dạng ký hiệu khoa học, nếu phần hệ số có ký số phía trước dấu chấm khác 0 thì ta gọi là biểu diễn theo dạng chuẩn tắc (*nomalize*).

Ví dụ:

- 4.27×10^5 là biểu diễn theo dạng chuẩn tắc.
- 0.427×10^6 hay 42.7×10^4 là biểu diễn không theo dạng chuẩn tắc.

Trong hệ nhị phân cũng tồn tại cách biểu diễn theo dạng ký hiệu khoa học chuẩn tắc. Lúc này, một số nhị phân biểu diễn ở dạng chuẩn tắc sẽ có dạng $1.xxxx \times 2^{yyyy}$. Trong đó $xxxx$ là các ký số nhị phân 0 và 1, còn $yyyy$ là các ký số trong hệ thập phân.

1.2 Biểu diễn số thực theo dạng chuẩn IEEE - 754

Máy tính chỉ có thể hiểu được hai ký tự là 0 và 1, cho nên việc biểu diễn số nhị phân theo dạng chuẩn tắc để đưa vào máy tính là không thể thực hiện được. Do vậy, vào năm 1985, việc biểu diễn số thực dấu chấm động được chuẩn hóa bởi tổ chức IEEE (*Institute of Electrical and Electronic Engineers*) và được gọi là chuẩn IEEE - 754. Chuẩn IEEE - 754 có thể biểu diễn ở nhiều kích thước khác nhau như 16 bit, 32 bit, 64 bit... Trong đó kích thước biểu diễn 32 bit được gọi là dạng chính xác đơn (*single precision*), còn kích thước biểu diễn 64 bit được gọi là dạng chính xác kép (*double precision*). Khi biểu diễn ở dạng chính xác đơn, kích thước biểu diễn 32 bit sẽ được chia ra lần lượt như sau:

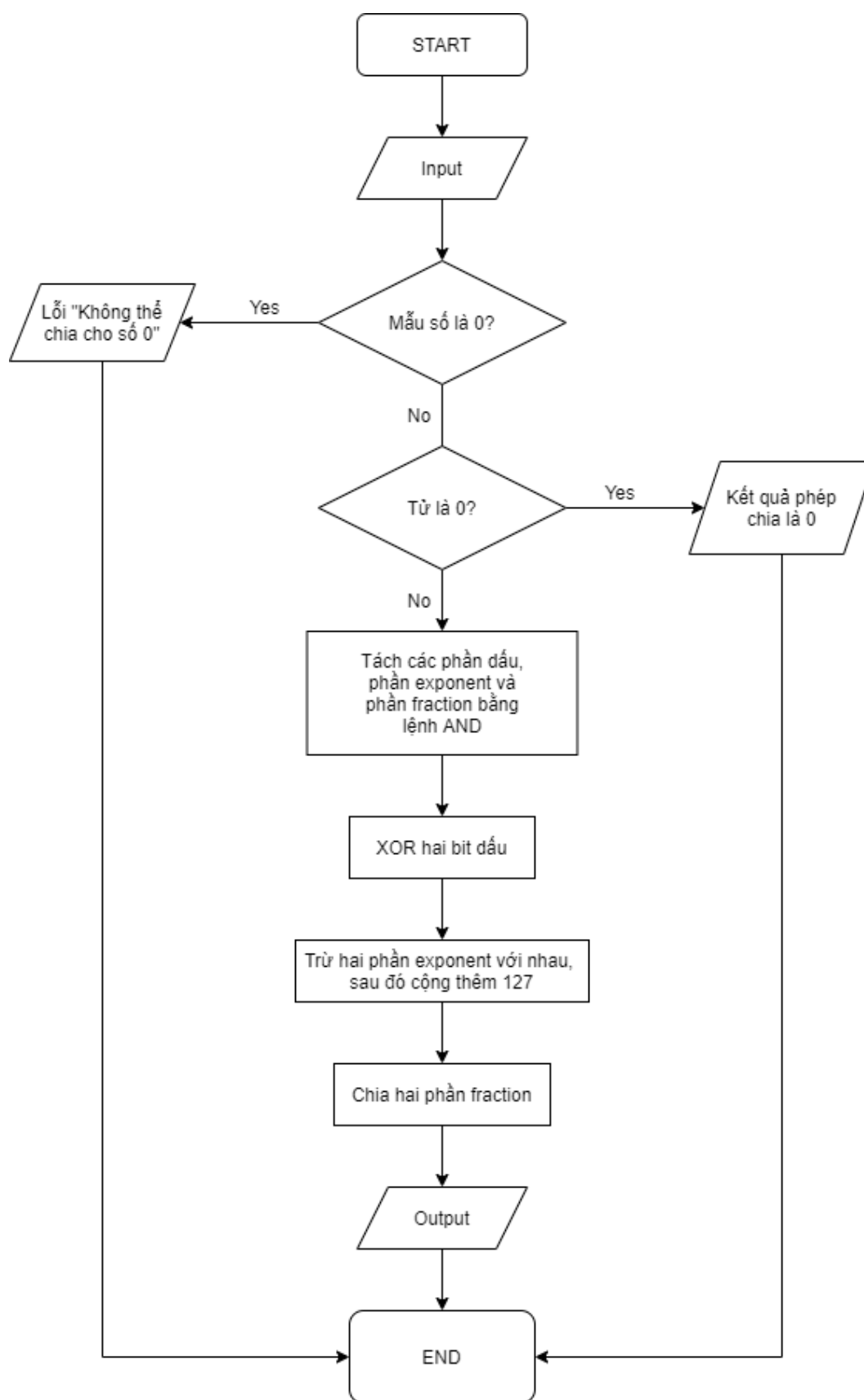
- Phần dấu (*sign*): bit có trọng số 31. Nếu là 0 thì biểu diễn số thực dương, còn nếu là 1 thì biểu diễn số thực âm.
- Phần mũ (*exponent*): bit có trọng số 23 - 30. Là phần mũ của số thực cộng thêm giá trị gọi là *bias*, đối với dạng chính xác đơn thì *bias* có giá trị là 127. Phần *bias* này dùng để đảm bảo rằng phần mũ luôn là một số dương.
- Phần phân số (*fraction*): bit có trọng số 0 - 22. Dùng để biểu diễn các ký số nằm phía sau dấu chấm trong phần hệ số (*coefficient*) của số nhị phân ở dạng chuẩn tắc.

2 Giải thuật chia hai số thực dấu chấm động

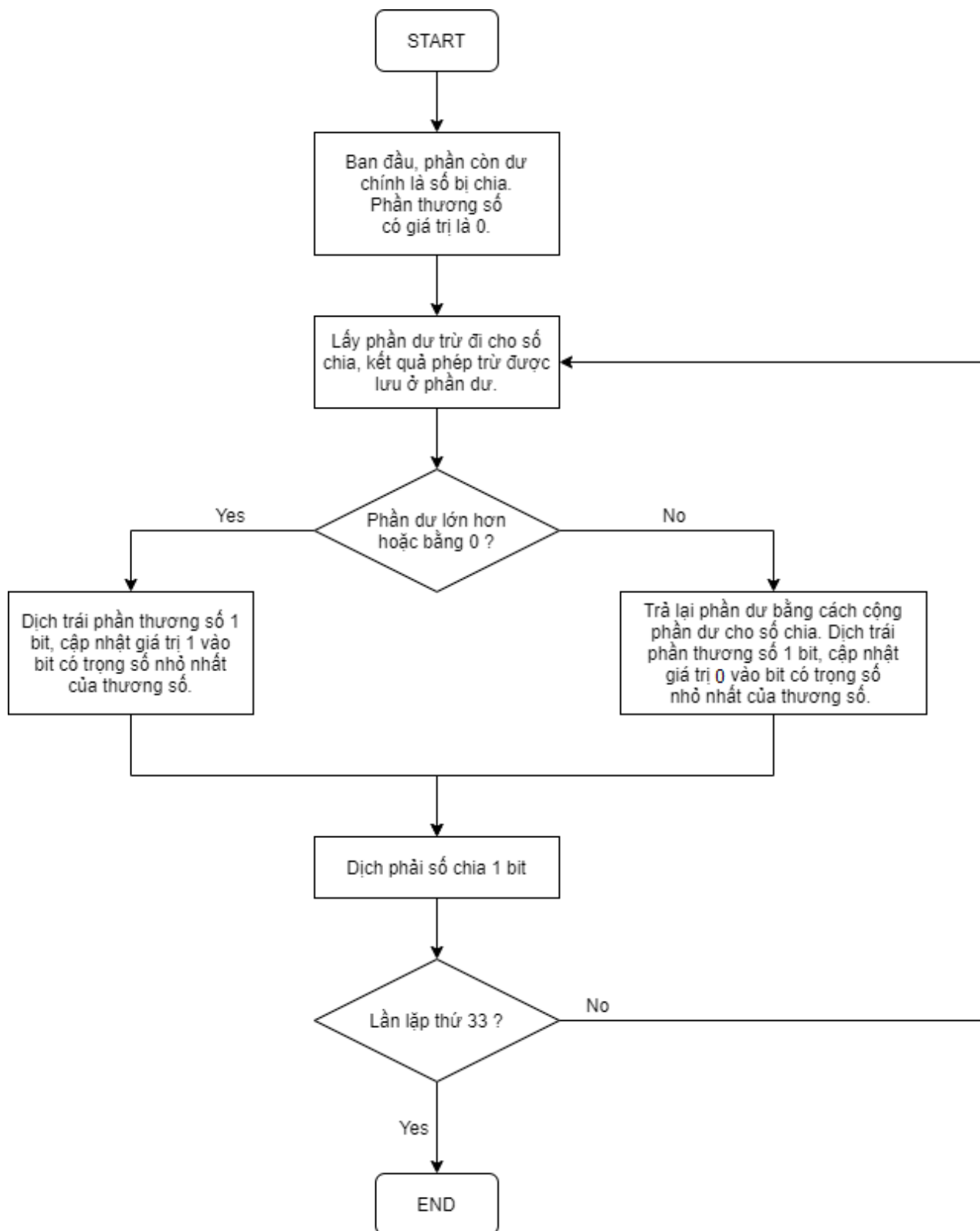
Để chia hai số thực dấu chấm động trong hợp ngữ MIPS, thông thường ta sẽ sử dụng lệnh *div.s*, tuy nhiên dưới đây sẽ trình bày phương pháp chia hai số thực dấu chấm động nhưng không sử dụng lệnh *div.s*, nghĩa là mô phỏng lại quá trình chia hai số thực dưới góc độ của phần cứng máy tính.

- Đầu tiên, sau khi nhập hai số thực vào máy tính, ta sẽ kiểm tra hai trường hợp đặc biệt, đó là trường hợp mẫu số bằng 0, trong trường hợp này chương trình sẽ xuất ra lỗi không thể chia cho 0, và tiếp theo là trường hợp tử số bằng 0 (lúc này mẫu số khác 0), trong trường hợp này ngay lập tức sẽ trả về kết quả của phép chia là 0.
- Nếu hai số thực nhập vào không thuộc hai trường hợp phía trên, ta sẽ thực hiện tách mỗi số thực được lưu trữ ở trong thanh ghi ra làm 3 phần, đó là phần dấu (*sign*), phần mũ *exponent* và phần phân số (*fraction*). Việc tách ra 3 phần sẽ được thực hiện bằng lệnh AND, và sau đó ta sẽ lần lượt thao tác trên từng phần
- Đối với phép chia, nếu hai số cùng dấu, thì kết quả phép chia sẽ là số dương, ngược lại, nếu hai số trái dấu, thì kết quả phép chia sẽ là số âm. Cho nên ta sẽ thực hiện XOR hai bit dấu của hai số thực với nhau.
- Tiếp theo, ta sẽ lấy phần mũ của số bị chia trừ đi cho số chia. Tuy nhiên, phép trừ này đã bao gồm việc triệt tiêu đi giá trị *bias*, là giá trị được dùng để đảm bảo phần mũ luôn là một số dương. Cho nên kết quả của phép trừ hai phần mũ sẽ được cộng thêm cho *bias* 127.
- Sau đó, ta sẽ chia hai phần thập phân của hai số thực. Đây là một quá trình phức tạp, sẽ được trình bày bằng sơ đồ giải thuật ở phía dưới.
- Kết quả của phép chia hai phần thập phân này sẽ được định dạng lại theo đúng chuẩn IEEE - 754. Nghĩa là, ta sẽ dịch trái kết quả của phép chia cho đến khi nào bit có trọng số cao nhất là 1, sau đó dịch trái thêm một lần nữa, mỗi lần dịch trái ta sẽ trừ đi exponent cho 1.
- Như vậy ta đã có được kết quả của phép chia theo đúng chuẩn IEEE - 754

Để trực quan hơn, ta sẽ mô phỏng lại giải thuật chia hai số thực dấu chấm động bằng sơ đồ thuật toán.



Hình 1: Sơ đồ tổng quan giải thuật chia hai số thực dấu chấm động



Hình 2: Sơ đồ quá trình chia phần thập phân của hai số thực

3 Kiểm thử chương trình

Giả sử CPI của tất cả các lệnh đều là 1 và tần số xung nhịp (*clock rate*) của CPU là 2.0 GHz. Khi đó:

$$\text{CPU TIME} = \frac{IC_R \cdot CPI + IC_I \cdot CPI + IC_J \times CPI}{\text{ClockRate}} = \frac{IC_R + IC_I + IC_J}{2.0 \cdot 10^9}$$

với IC_R, IC_I, IC_J lần lượt là số lệnh cần phải thực hiện của các loại lệnh dạng R, I, J .

Ví dụ	Số bị chia	Số chia	Kết quả	R-Type	I-Type	J-Type	CPU TIME
1	5.78	3.5	1.6514	147	133	27	1.53×10^{-7}
2	-7.85	3.664	-2.142467	142	131	26	1.495×10^{-7}
3	57.223	3.2225	17.75733	141	132	26	1.495×10^{-7}
4	53.89	-2.5644	-21.014662	143	130	26	1.495×10^{-7}
5	0	2.67	0.0	14	17	2	1.65×10^{-8}
6	2.67	0	khong the chia cho 0	9	14	1	1.2×10^{-8}
7	0	0	khong the chia cho 0	9	14	1	1.2×10^{-8}