Bộ chia

1. Các lệnh chia trong RISC-V (RV32M)

| **Lệnh** | **Toán hạng** | **Kết quả** | **Ý nghĩa** |
| --- | --- | --- | --- |
| DIV | signed ÷ signed | quotient | Thương có dấu |
| DIVU | unsigned ÷ unsigned | quotient | Thương không dấu |
| REM | signed ÷ signed | remainder | Dư có dấu |
| REMU | unsigned ÷ unsigned | remainder | Dư không dấu |

DIV 7 ÷ 3 = 2

REM 7 ÷ 3 = 1

DIV -7 ÷ 3 = -2

REM -7 ÷ 3 = -1

1. Vấn đề cần giải quyết trong phần cứng

CPU làm việc với **số nhị phân** (binary).

Phép chia không thể thực hiện trong 1 chu kỳ như cộng/trừ → cần thuật toán lặp.

Cần xử lý **có dấu / không dấu**.

Cần định nghĩa rõ trường hợp đặc biệt:

* Chia cho 0 → trong RISC-V, quotient = -1 (tất cả bit = 1), remainder = dividend.
* Chia tràn (ví dụ -2^31 ÷ -1) → quotient = -2^31, remainder = 0.

1. Nguyên lý cơn bản của chia nhị phân

Giống như phép chia tay truyền thống, phần cứng dùng **thuật toán trừ dịch (shift-subtract division algorithm)**:

* 1. **Chuẩn bị**
  + Dividend (số bị chia).
  + Divisor (số chia).
  + Quotient = 0.
  + Remainder = dividend (ban đầu).
  + Dùng mặt nạ bit q\_mask để biết đang set bit nào trong quotient.
  1. **Lặp 32 lần (với số 32 bit)**

So sánh: divisor ≤ remainder?

Nếu có: remainder = remainder – divisor, quotient = quotient | q\_mask.

Dịch divisor sang phải 1 bit.

Dịch q\_mask sang phải 1 bit.

* 1. **Kết quả**

Quotient (thương) nằm trong quotient.

Remainder (dư) nằm trong remainder.

* Đây là lý do trong code của bạn có divisor\_q 63 bit (dịch trái ban đầu để canh bit cao), và q\_mask\_q khởi tạo = 0x8000\_0000.