



COMPUTER ENGINEERING

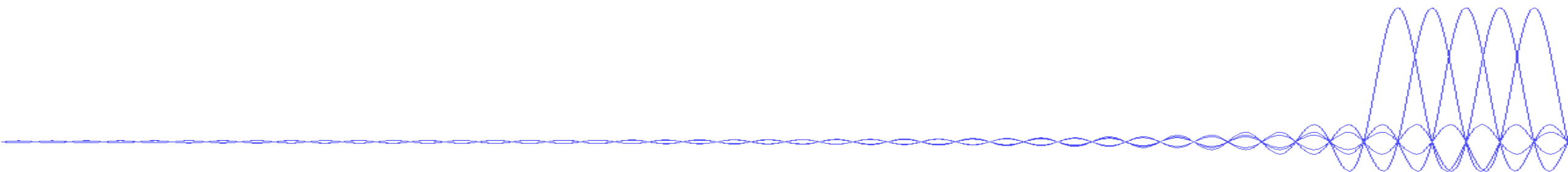
# KIẾN TRÚC MÁY TÍNH



**UIT**  
TRƯỜNG ĐẠI HỌC  
CÔNG NGHỆ THÔNG TIN

## Tuần 8-9

# CÁC PHÉP TOÁN SỐ HỌC BÊN TRONG MÁY TÍNH





# Phép toán số học bên trong máy tính

## Mục tiêu:

Hiểu được cách thức máy tính thực thi các phép toán số học sau:

- ✓ Phép cộng
- ✓ Phép trừ
- ✓ Phép nhân
- ✓ Phép chia

Tài liệu tham khảo:

***Computer Organization and Design: The Hardware/Software Interface***,  
Patterson, D. A., and J. L. Hennessy, Morgan Kaufman, Revised Fourth Edition,  
2011.



# Phép toán số học bên trong máy tính

1. Phép cộng và phép trừ
2. Phép nhân
3. Phép chia



# Phép cộng và phép trừ

## ■ Phép cộng

□ Ví dụ:  $7 + 6$  (32 bit)

$$\begin{array}{r}
 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0111 \\
 + 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0110 \\
 \hline
 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 1101
 \end{array}$$

A	B	Tổng	Nhớ
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

□ Tổng quan:  $a_n a_{n-1} \dots a_1 a_0 + b_n b_{n-1} \dots b_1 b_0$

■ Cộng tuần tự các bit từ trọng số thấp đến các bit có trọng số cao

■ Các bit nhớ từ phép cộng bit có trọng số thấp sẽ được cộng dồn vào phép cộng bit có trọng số cao liền kề

$$\begin{array}{r}
 \text{+} \quad \text{Nhớ}_{in} \quad \dots \quad \text{0} \quad \text{0} \quad \text{1} \quad \text{1} \quad \text{1} \\
 \text{+} \quad A \quad \dots \quad \text{0} \quad \text{0} \quad \text{1} \quad \text{1} \quad \text{1} \\
 \text{+} \quad B \quad \dots \quad \text{0} \quad \text{0} \quad \text{1} \quad \text{1} \quad \text{0} \\
 \hline
 \text{Tổng} \quad \dots \quad \text{0} \quad \text{1} \quad \text{1} \quad \text{0} \quad \text{1} \\
 \text{Nhớ}_{out} \quad \dots \quad \text{0} \quad \text{0} \quad \text{1} \quad \text{1} \quad \text{0}
 \end{array}$$



# Phép cộng và phép trừ

## ■ Phép trừ

□ Ví dụ: 6 - 3 (32 bit)

$$\begin{array}{r}
 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0110 \\
 - 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0011 \\
 \hline
 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0011
 \end{array}$$

A	B	Hiệu	Mượn
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

□ Tổng quan:  $a_n a_{n-1} \dots a_1 a_0 - b_n b_{n-1} \dots b_1 b_0$

■ Trừ tuần tự các bit từ trọng số thấp đến các bit có trọng số cao

■ Các bit mượn từ phép trừ bit có trọng số thấp sẽ được trừ dồn vào phép trừ bit có trọng số cao liền kề

$$\begin{array}{rcl}
 A & \dots & 0\ 0\ 1\ 1\ 0 \\
 B & \dots & 0\ 0\ 0\ 1\ 1 \\
 \hline
 \text{Trả} & \dots & \\
 \text{Hiệu} & \dots & 0\ 0\ 0\ 1\ 1 \\
 \text{Mượn} & \dots & 0\ 0\ 0\ 1\ 1
 \end{array}$$

*Note: Red dashed arrows in the original image point from the borrow bits (1s) of the lower positions to the corresponding bits of the higher positions in the 'Hiệu' and 'Mượn' rows.*



# Phép cộng và phép trừ

## ■ Phép trừ

□ Ta có thể thực hiện phép trừ bằng phép biến đổi như sau:

$$A - B = A + (-B) = A + \text{Bù 2}(B) = A + \text{Bù 1}(B) + 1$$

□ Tổng quan:  $a_n a_{n-1} \dots a_1 a_0 - b_n b_{n-1} \dots b_1 b_0$

■ Biểu diễn giá trị  $-B$  ở dạng số bù 2

■ Thực hiện phép cộng

A	B	Hiệu	Mượn
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$\begin{array}{r}
 \begin{array}{c} \text{---} \\ \text{---} \end{array} \begin{array}{l} A \\ B \end{array} \begin{array}{l} \dots 0 \ 0 \ 1 \ 1 \ 0 \\ \dots 0 \ 0 \ 0 \ 1 \ 1 \end{array} \\
 \longleftrightarrow \\
 \begin{array}{r}
 \begin{array}{c} \text{+} \\ \text{+} \end{array} \begin{array}{l} A \\ \text{Bù 1}(B) \\ 1 \end{array} \begin{array}{l} \dots 0 \ 0 \ 1 \ 1 \ 0 \\ \dots 1 \ 1 \ 1 \ 0 \ 0 \\ \dots 0 \ 0 \ 0 \ 0 \ 1 \\ \hline \dots 0 \ 0 \ 0 \ 1 \ 1 \end{array}
 \end{array}$$



# Phép cộng và phép trừ

■ **Tràn (Overflow):** là khi kết quả của phép toán vượt quá số bit biểu diễn phần giá trị

□  $n$  bit biểu diễn được số có dấu bù 2 từ  $-2^{n-1}$  đến  $+2^{n-1}-1$

□ Overflow luôn cho ra 1 kết quả sai

□ Ví dụ: Thực hiện phép cộng hai số có dấu 5 bit ( $8 + 9$ ), kết quả lưu trong 5 bit

$$\begin{array}{r} + \quad A \quad 0 \ 1 \ 0 \ 0 \ 0 \\ \quad B \quad 0 \ 1 \ 0 \ 0 \ 1 \\ \hline \text{Tổng} \quad 1 \ 0 \ 0 \ 0 \ 1 \end{array}$$

→ Như vậy, kết quả phép toán sai (tổng của hai số dương là một số âm), nguyên nhân là do kết quả thực sự của phép toán ( $8 + 9 = 17$ ) nằm ngoài phạm vi biểu diễn số có dấu bù 2 có 5 bit, dẫn đến hiện tượng tràn số học

□ **Vấn đề:** Làm sao máy tính phát hiện được phép toán thực hiện có bị tràn hay không?



# Phép cộng và phép trừ

## ■ Phát hiện tràn (Overflow)

□ Ví dụ: Thực hiện phép cộng/trừ số 4 bit có dấu bù 2

$$\begin{array}{r} (+7) \quad 0111 \\ + (+2) \quad 0010 \\ \hline (+9) \quad 1001 \\ c_4 = 0 \\ c_3 = 1 \end{array}$$

$$\begin{array}{r} (+7) \quad 0111 \\ + (-2) \quad 1110 \\ \hline (+5) \quad 10101 \\ c_4 = 1 \\ c_3 = 1 \end{array}$$

$$\begin{array}{r} (-7) \quad 1001 \\ + (+2) \quad 0010 \\ \hline (-5) \quad 1011 \\ c_4 = 0 \\ c_3 = 0 \end{array}$$

$$\begin{array}{r} (-7) \quad 1001 \\ + (-2) \quad 1110 \\ \hline (-9) \quad 10111 \\ c_4 = 1 \\ c_3 = 0 \end{array}$$

□ Cờ tràn (Overflow) =  $c_3 \oplus c_4$





# Phép cộng và phép trừ

## ■ Phát hiện tràn (Overflow)

- Một cách tổng quát khi thực hiện phép cộng/trừ số  $n$  bit có dấu bù 2
- Cờ tràn (Overflow) =  $c_{n-1} \oplus c_n$
- Mạch cộng/ trừ có thể bổ sung mạch kiểm tra tràn với 1 cổng XOR.
- Nếu sau khi thực hiện phép tính, cờ tràn có giá trị bằng “1” thì ta không cần quan tâm giá trị của phép tính vì giá trị đó bị sai



# Phép toán số học bên trong máy tính

1. Phép cộng và phép trừ
2. Phép nhân
3. Phép chia



# Phép nhân

## ■ Ví dụ: $8 \times 9 = 72$

$$\begin{array}{r} \text{Số bị nhân:} \quad \quad \quad \times \quad 1000 \quad (8) \\ \text{Số nhân:} \quad \quad \quad \quad \quad 1001 \quad (9) \\ \hline \quad \quad \quad \quad \quad 1000 \\ \quad \quad \quad + \quad \quad \quad 0000 \\ \quad \quad \quad \quad \quad 0000 \\ \quad \quad \quad \quad \quad \quad 1000 \\ \hline \text{Tích:} \quad \quad \quad 1001000 \quad (72) \end{array}$$

## ■ Phương pháp 1

- Nhân số bị nhân với từng bit của số nhân từ phải sang trái
- Nếu bit số nhân bằng 1 thì kết quả bằng chính số bị nhân
- Nếu bit số nhân bằng 0 thì kết quả bằng 0
- Kết quả của phép nhân bit sau sẽ dịch trái 1 bit so với kết quả của phép nhân bit trước đó
- Cộng tất cả các kết quả phép nhân sẽ được giá trị tích cuối cùng



# Phép nhân

## ■ Ví dụ: $8 \times 9$ (1000 x 1001)

Số bị nhân: 10000000

Số nhân: 0000

Tích: 01001000

## ■ Phương pháp 2

### □ 0. Khởi tạo

- Thanh ghi Số bị nhân 8 bit, với 4 bit thấp chứa Số bị nhân và 4 bit cao chứa giá trị 0
- Thanh ghi Số nhân 4 bit
- Thanh ghi Tích 8 bit lưu giá trị 0

### □ 1. Kiểm tra bit [0] của Số nhân

- Nếu bằng 1, Tích = Tích + Số bị nhân
- Nếu bằng 0, Tích giữ nguyên

### □ 2. Dịch trái Số bị nhân

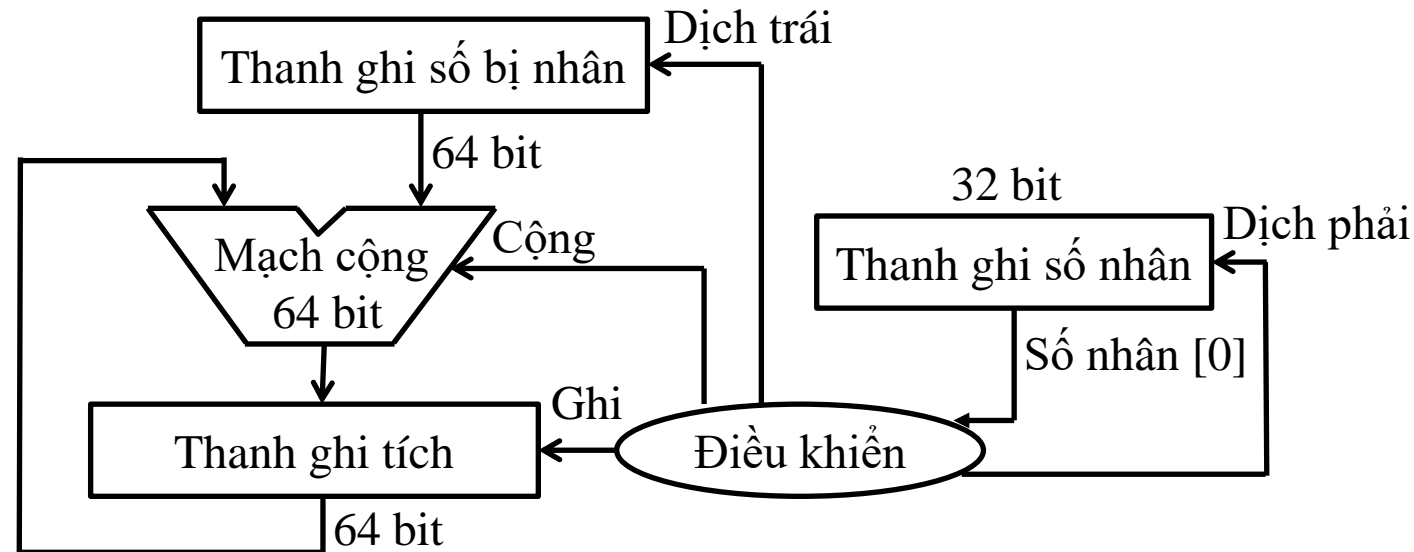
### □ 3. Dịch phải Số nhân

### □ Lặp lại bước 1-3 cho đủ 4 lần nữa (một cách tổng quát, lặp lại $n$ lần đối với phép nhân 2 số $n$ bit)



# Phép nhân

## ■ Kiến trúc hiện thực phần cứng phép nhân 32 bit bên trong máy tính



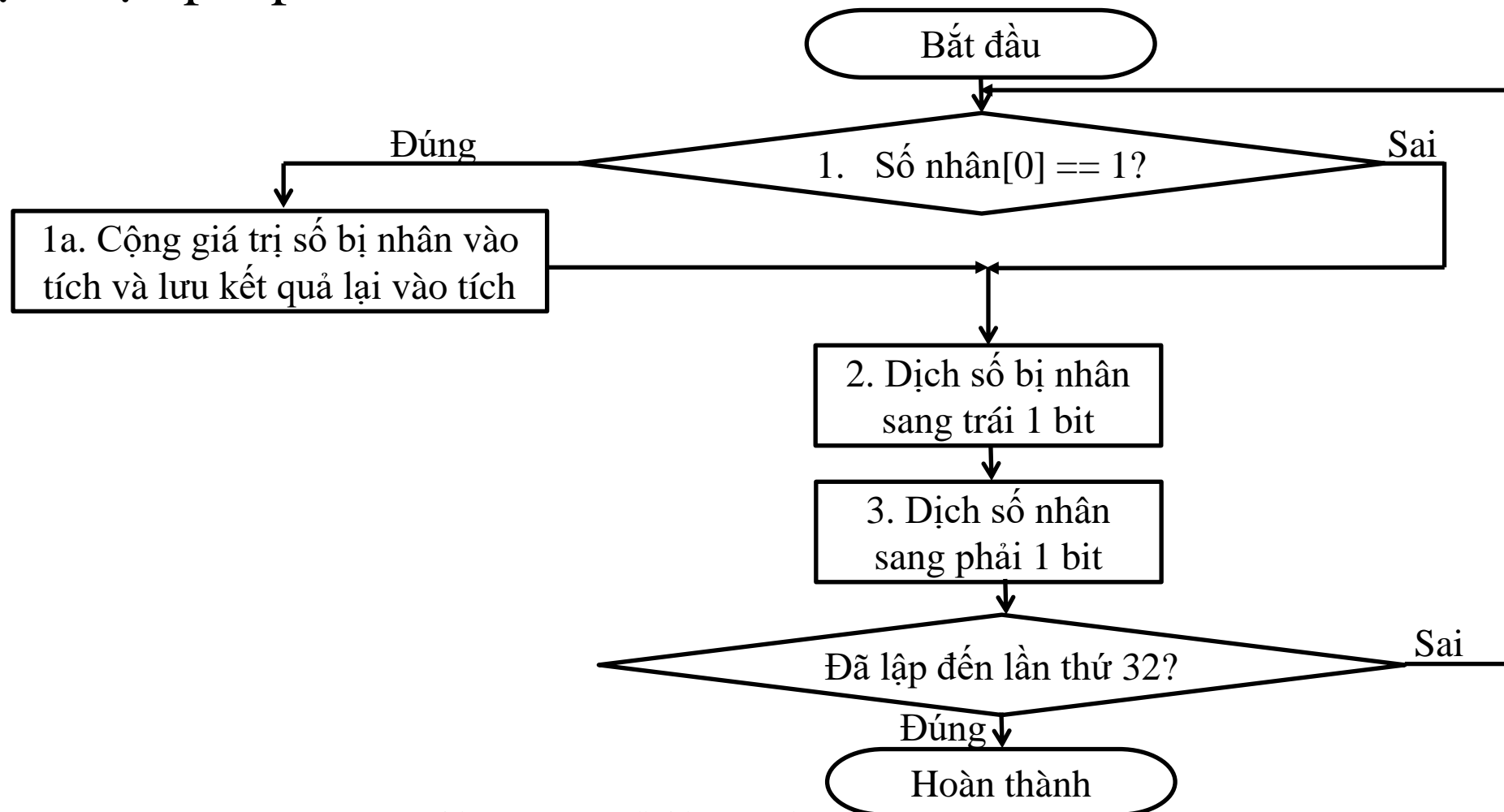
## ■ Khởi tạo giá trị ban đầu cho các thanh ghi:

- ❑ 32 bit thấp của thanh ghi số bị nhân lưu giá trị số bị nhân, 32 bit cao lưu giá trị 0
- ❑ 32 bit của thanh ghi số nhân lưu giá trị số nhân
- ❑ 64 bit của thanh ghi tích lưu giá trị 0



# Phép nhân

## ■ Giải thuật thực hiện phép nhân hai số 32 bit

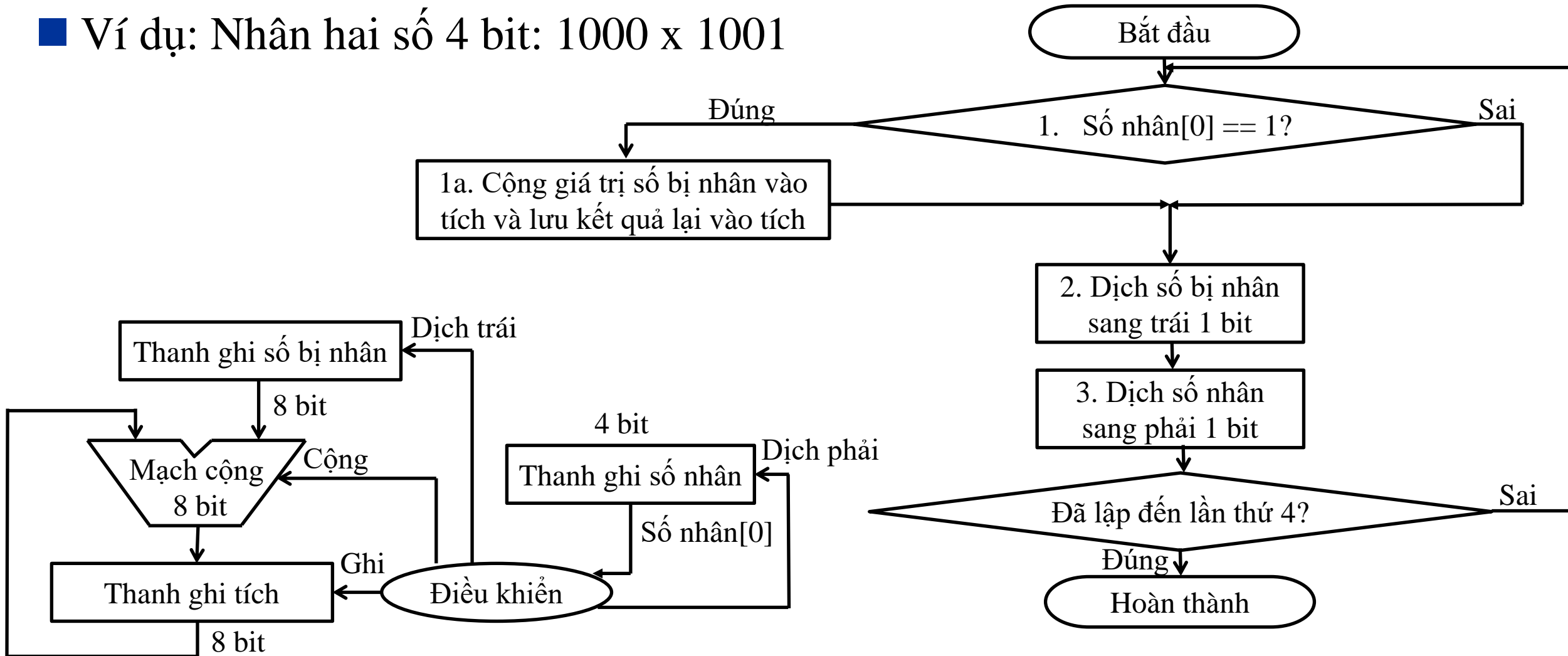




# Phép nhân

COMPUTER ENGINEERING

■ Ví dụ: Nhân hai số 4 bit: 1000 x 1001





# Phép nhân

Lần lập	Thực hiện	Số nhân	Số bị nhân	Tích
0	Khởi tạo giá trị	1001	0000 1000	0000 0000
1	1a: 1 → Tích = Tích + Số bị nhân	1001	0000 1000	0000 1000
	2: dịch số bị nhân sang trái 1 bit	1001	0001 0000	0000 1000
	3: dịch số nhân sang phải 1 bit	0100	0001 0000	0000 1000
2	1a: 0 → Tích giữ nguyên giá trị	0100	0001 0000	0000 1000
	2: dịch số bị nhân sang trái 1 bit	0100	0010 0000	0000 1000
	3: dịch số nhân sang phải 1 bit	0010	0010 0000	0000 1000
3	1: 0 → Tích giữ nguyên giá trị	0010	0010 0000	0000 1000
	2: dịch số bị nhân sang trái 1 bit	0010	0100 0000	0000 1000
	3: dịch số nhân sang phải 1 bit	0001	0100 0000	0000 1000
4	1: 1 → Tích = Tích + Số bị nhân	0001	0100 0000	0100 1000
	2: dịch số bị nhân sang trái 1 bit	0001	1000 0000	0100 1000
	3: dịch số nhân sang phải 1 bit	0000	1000 0000	0100 1000 = 72





## ■ Phép nhân hai số có dấu

□ Cách đơn giản để thực hiện phép nhân có dấu là tách phần trị tuyệt đối và dấu của số bị nhân và số nhân ra

- Lấy phần trị tuyệt đối dương tương ứng của số nhân và số bị nhân nhân nhau
- Sau đó xét dấu cho tích dựa vào dấu của số nhân và số bị nhân (có thể dùng phép XOR)



## ■ Phép nhân trong tập lệnh MIPS

□ Hai thanh ghi 32 bit được sử dụng để lưu giá trị tích trong phép nhân

■ HI: Lưu 32 bit trọng số cao của giá trị tích

■ LO: Lưu 32 bit trọng số thấp của giá trị tích

□ Sử dụng tập lệnh MIPS thực hiện phép nhân

■ `mult rs, rt` / `multu rs, rt`

□ Giá trị tích 64 bit được lưu mặc định trong 2 thanh ghi HI/LO

■ `mfhi rd` / `mflo rd`

□ Lấy kết quả từ thanh ghi HI/LO ra thanh ghi rd

■ `mul rd, rs, rt`

□ 32 bit trọng số thấp của giá trị tích được ghi vào thanh ghi rd



# Phép toán số học bên trong máy tính

1. Phép cộng và phép trừ
2. Phép nhân
3. Phép chia



# Phép chia

## ■ Ví dụ: 40:19

### □ Giải thuật 1

$$\begin{array}{r} \text{Số bị chia} \quad \text{Số chia} \\ \downarrow \quad \downarrow \\ 101000 \mid 10011 \\ -10011 \quad 10 \swarrow \\ \hline 01 \quad \text{Thương} \\ 010 \\ \downarrow \\ \text{Số dư} \end{array}$$

## ■ Phương pháp 1:

□ Kiểm tra Số chia có bằng 0 hay không

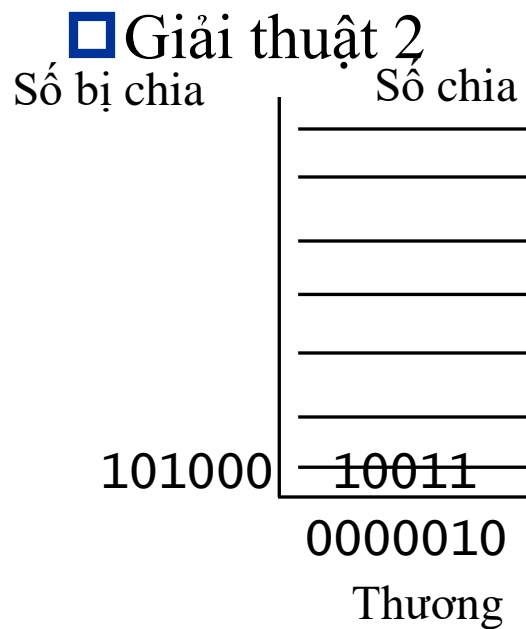
□ Khi Số chia khác 0

- Nếu Số chia  $\leq$  Các bit của Số bị chia, đặt “1” vào Thương, lấy Số bị chia – Số chia
- Ngược lại, đặt “0” vào Thương, thêm bit kế tiếp của Số bị chia



# Phép chia

## ■ Ví dụ: 40:19



## ■ Phương pháp 2:

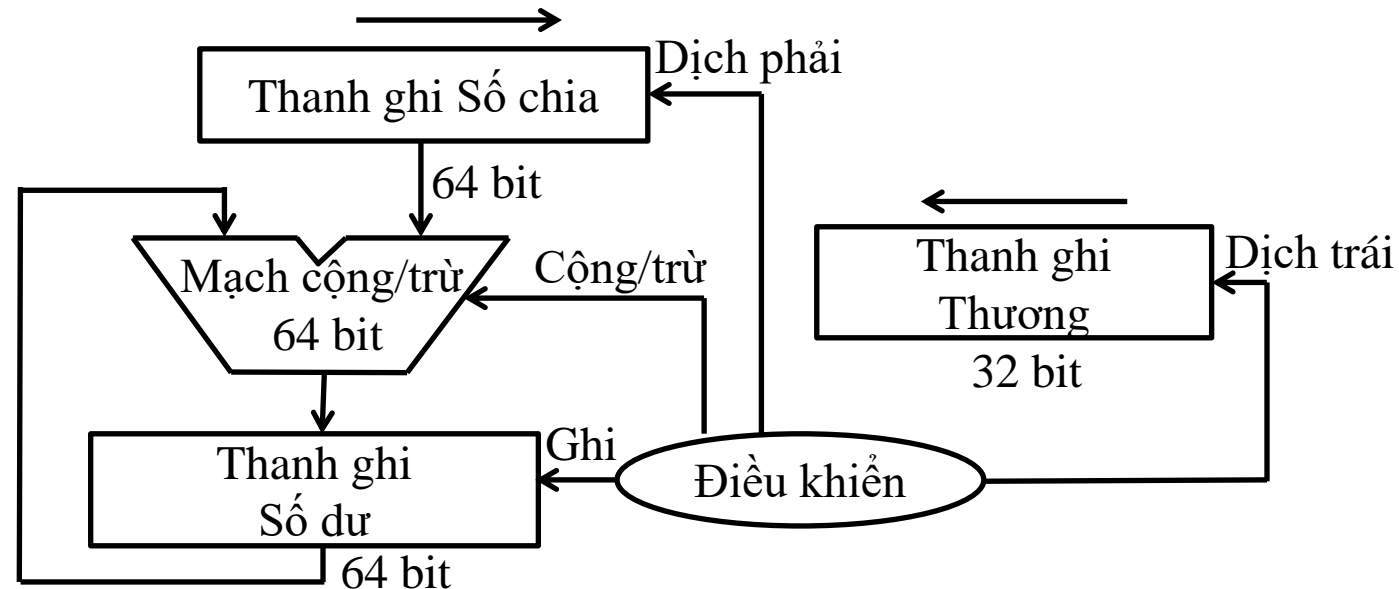
- Kiểm tra Số chia có bằng 0 hay không
- Khi Số chia khác 0
  - 1a. Đặt Số bị chia vào 12 bit với 6 bit trọng số cao chứa các bit 0
  - 1b. Đặt Số chia vào 12 bit với 6 bit trọng số thấp chứa các bit 0
  - 2. Nếu Số chia  $\leq$  Các bit của Số bị chia
    - Đặt “1” vào Thương, lấy Số bị chia – Số chia để được Số bị chia mới
  - Ngược lại, Số chia  $>$  Các bit của Số bị chia
    - Đặt “0” vào Thương, giữ nguyên số bị chia
  - 3. Dịch phải Số chia 1 bit
  - Lặp lại bước 2 và 3 thêm 6 lần.



# Phép chia

COMPUTER ENGINEERING

## ■ Kiến trúc hiện thực phần cứng phép chia 32 bit bên trong máy tính



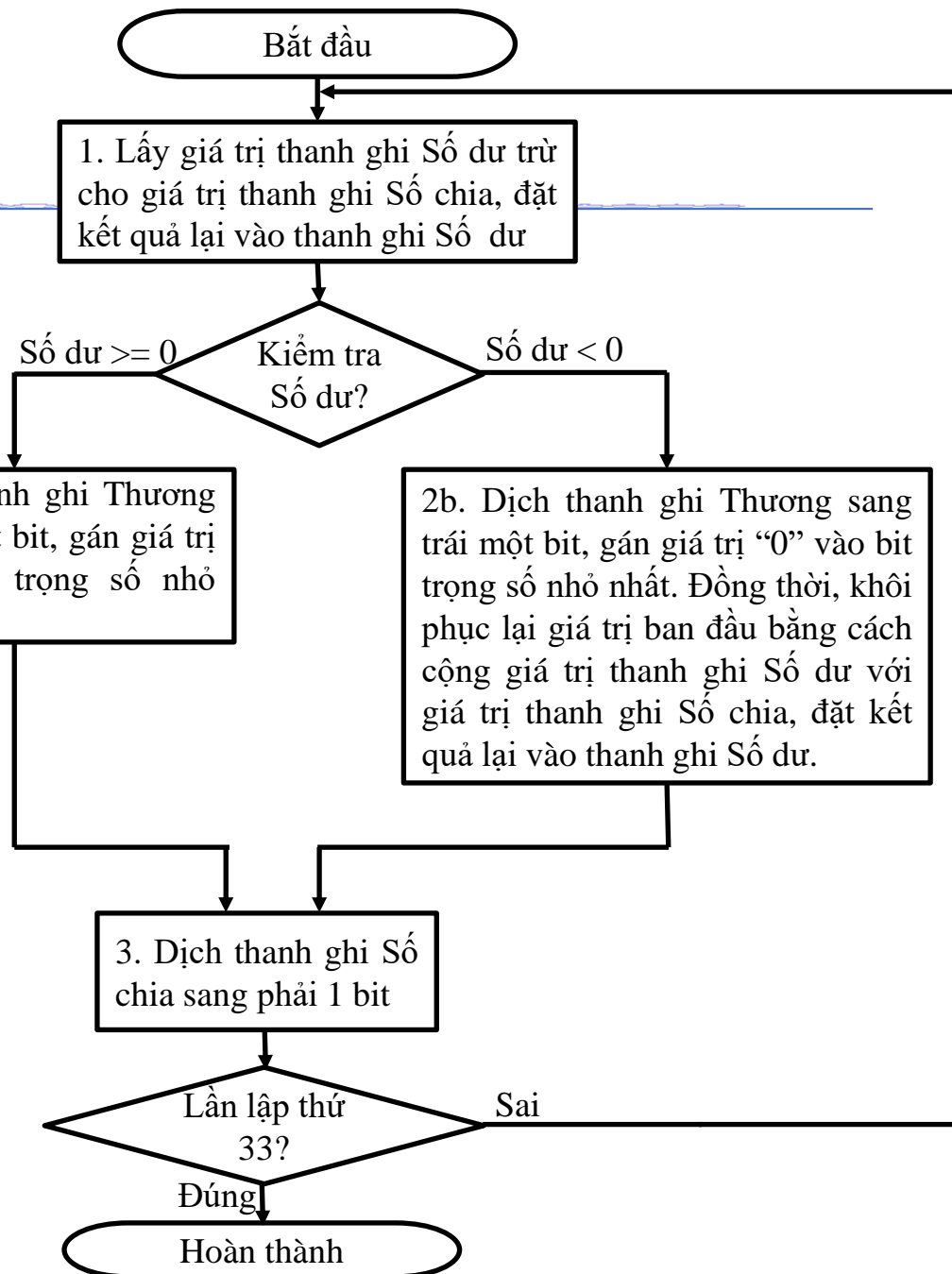
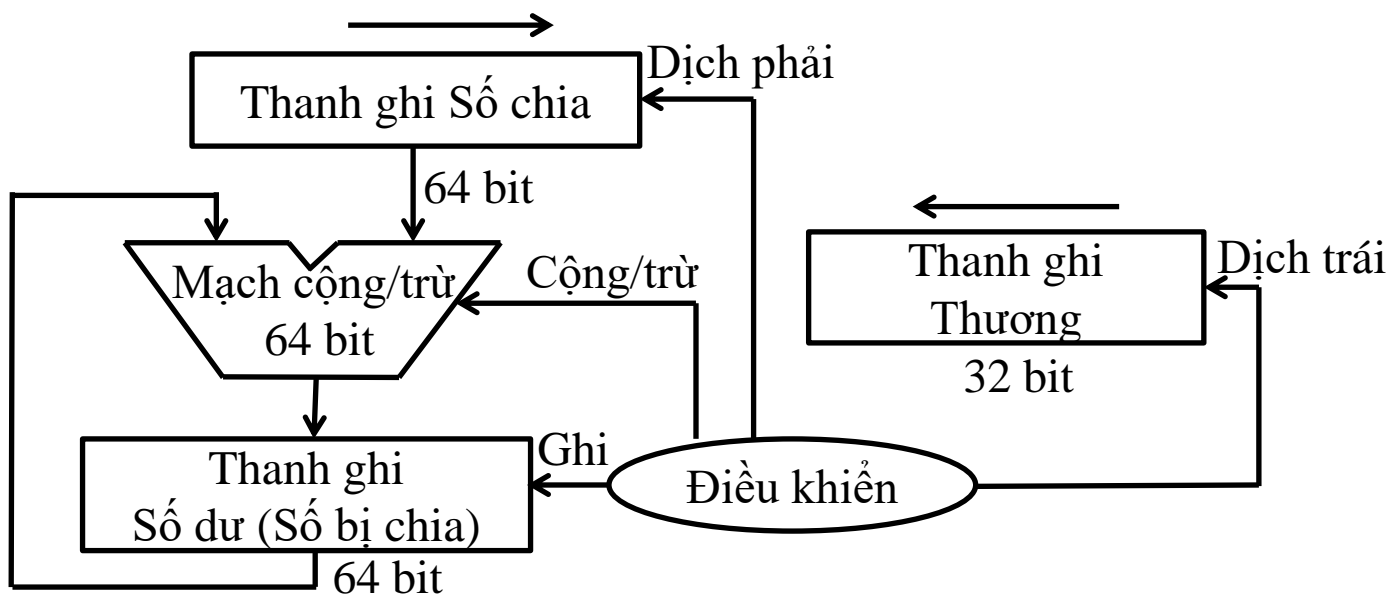
## ■ Khởi tạo giá trị ban đầu cho các thanh ghi:

- ❑ 32 bit thấp của thanh ghi Số dư lưu giá trị Số bị chia, 32 bit cao lưu giá trị 0
- ❑ 32 bit thấp của thanh ghi Số chia lưu giá trị 0, 32 bit cao lưu giá trị Số chia
- ❑ 32 bit của thanh ghi Thương lưu giá trị 0



# Phép chia

## ■ Giải thuật thực hiện phép chia hai số 32 bit

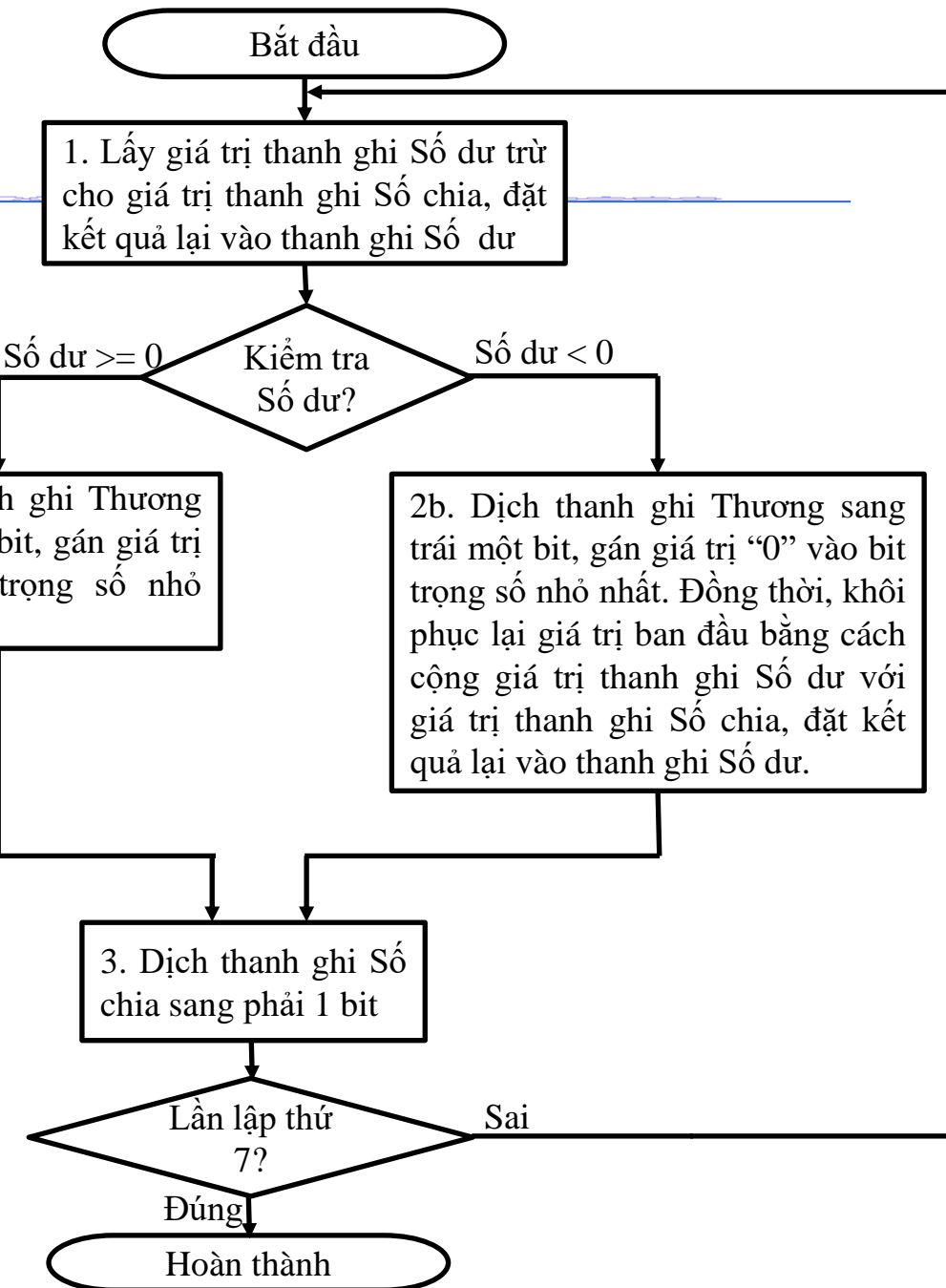
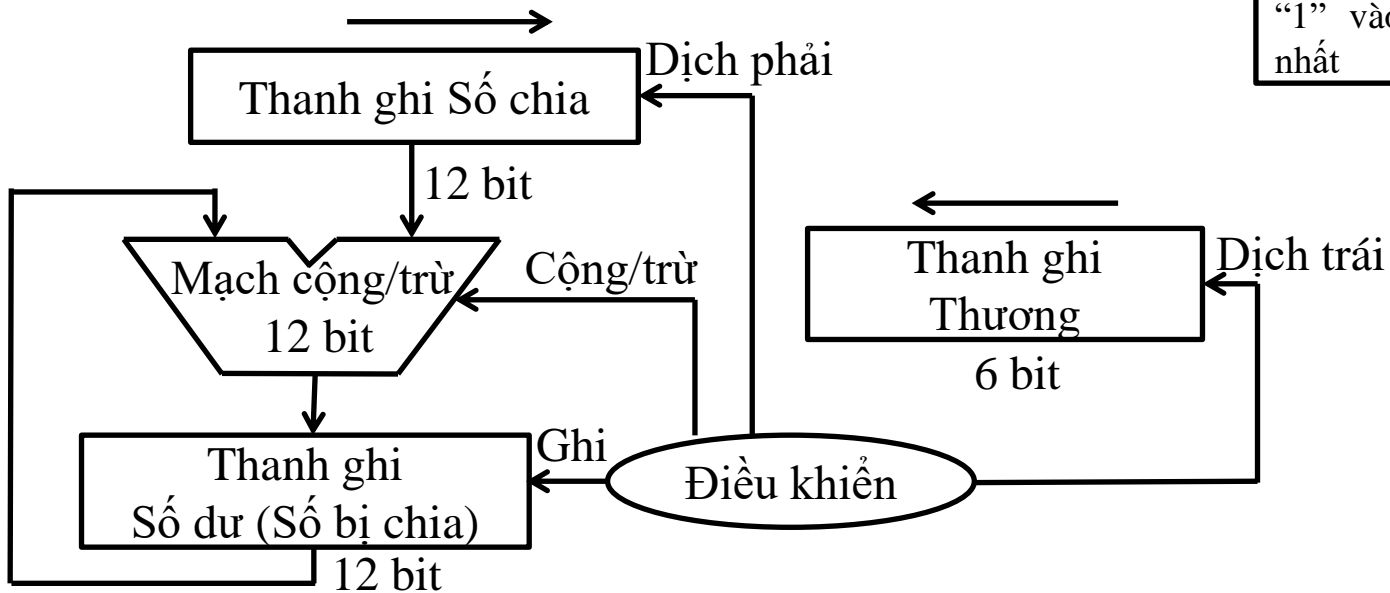




# Số chia

## ■ Ví dụ:

■ Chia hai số 6 bit: 40:19 (101000 : 010011)

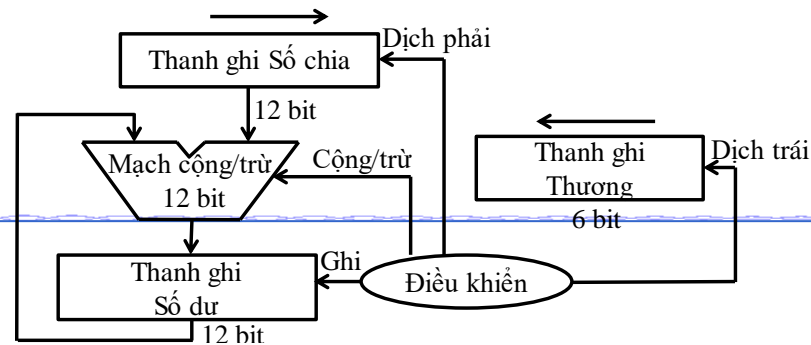
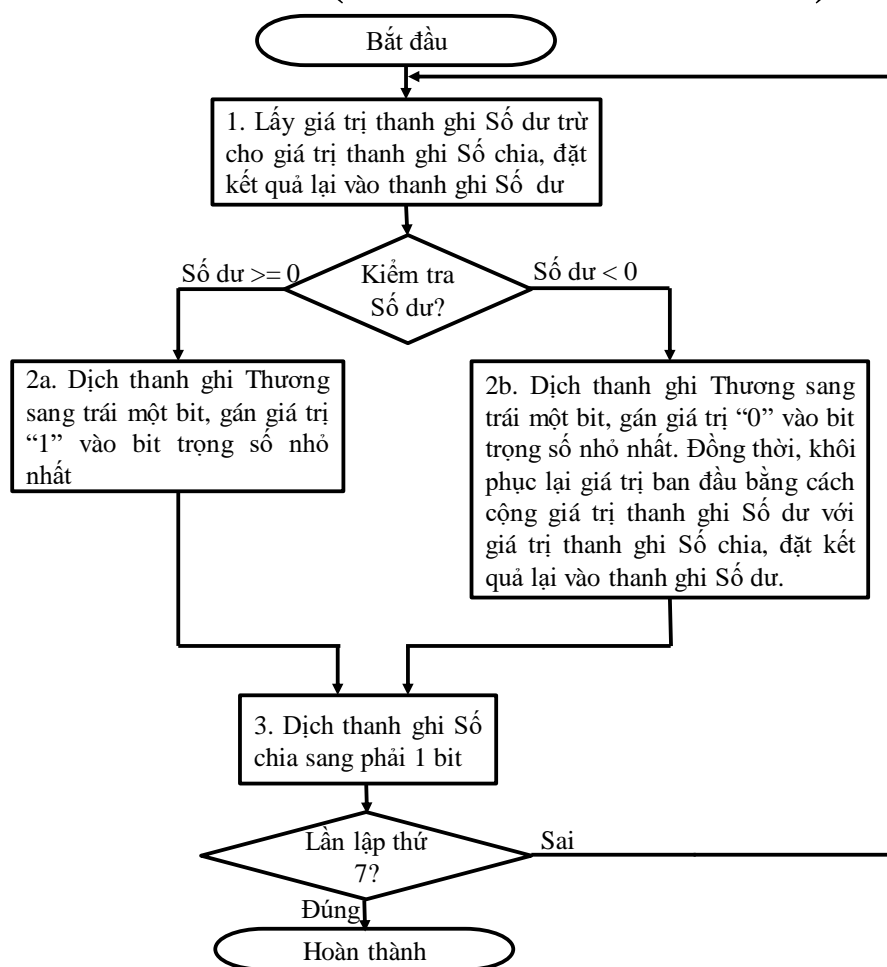






# Số chia

■ 40:19 (101000 : 10011)



Lặp	Thực thi	Thương (Q)	Số chia (D)	Số dư (R)
0	Khởi tạo giá trị	000 000	010 011 000 000	000 000 101 000
1	$R = R - D$	000 000	010 011 000 000	101 101 101 000
	$R < 0, R = R + D$ , dịch trái Q 1 bit	000 000	010 011 000 000	000 000 101 000
	Dịch phải D 1 bit	000 000	001 001 100 000	000 000 101 000
2	$R = R - D$	000 000	001 001 100 000	110 111 001 000
	$R < 0, R = R + D$ , dịch trái Q 1 bit	000 000	001 001 100 000	000 000 101 000
	Dịch phải D 1 bit	000 000	000 100 110 000	000 000 101 000
3	$R = R - D$	000 000	000 100 110 000	111 011 111 000
	$R < 0, R = R + D$ , dịch trái Q 1 bit	000 000	000 100 110 000	000 000 101 000
	Dịch phải D 1 bit	000 000	000 010 011 000	000 000 101 000
4	$R = R - D$	000 000	000 010 011 000	111 110 010 000
	$R < 0, R = R + D$ , dịch trái Q 1 bit	000 000	000 010 011 000	000 000 101 000
	Dịch phải D 1 bit	000 000	000 001 001 100	000 000 101 000
5	$R = R - D$	000 000	000 001 001 100	111 110 111 100
	$R < 0, R = R + D$ , dịch trái Q 1 bit	000 000	000 001 001 100	000 000 101 000
	Dịch phải D 1 bit	000 000	000 000 100 110	000 000 101 000
6	$R = R - D$	000 000	000 000 100 110	000 000 000 010
	$R > 0$ , dịch trái Q 1 bit, $Q_0 = 1$	000 001	000 000 100 110	000 000 000 010
	Dịch phải D 1 bit	000 001	000 000 010 011	000 000 000 010
7	$R = R - D$	000 001	000 000 010 011	111 111 101 111
	$R < 0, R = R + D$ , dịch trái Q 1 bit	000 010	000 000 010 011	000 000 000 010
	Dịch phải D 1 bit	000 010	000 000 001 001	000 000 000 010



## ■ Phép chia hai số có dấu

□ Bước 1: Bỏ qua bit dấu, thực hiện phép chia như bình thường .

□ Bước 2:

■ Dấu của thương được đảo nếu dấu của Số bị chia và Số chia trái nhau

■ Số dư = Số bị chia – (Thương x Số chia)



## ■ Phép chia trong tập lệnh MIPS

- Hai thanh ghi 32 bit được sử dụng để lưu giá trị thương và số dư trong phép chia
  - HI: Lưu 32 bit trọng số cao của giá trị số dư
  - LO: Lưu 32 bit trọng số thấp của giá trị thương
- Sử dụng tập lệnh MIPS thực hiện phép chia
  - `div rs, rt` / `divu rs, rt`
    - Giá trị số dư 32 bit và thương 32 bit được lưu mặc định trong 2 thanh ghi HI/LO
  - `mfhi rd` / `mflo rd`
    - Lấy kết quả từ thanh ghi HI/LO ra thanh ghi rd



# Tổng kết

- Hiểu quy tắc thực hiện các phép toán số học (cộng, trừ, nhân, chia) trên số nguyên trong máy tính
- Hiểu cách thiết kế mạch nhân, chia cơ bản cho số nguyên trong máy tính