TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN TOÁN ỨNG DỤNG & TIN HỌC

Bài giảng

KIẾN TRÚC MÁY TÍNH

Giảng viên : Phạm Huyền Linh
Bộ môn : Toán Tin

CHƯƠNG 3

SỐ HỌC MÁY TÍNH

(Computer Architecture)

Giảng viên: Phạm Huyền Linh

Chương 3

3.1. Biểu diễn số nguyên (Integer Representation)

3.2. Các phép tính với số nguyên (Integer Arithmetic)

3.3. Biểu diễn số dấu phẩy động (Floating-Point Representation)

3.4. Các phép tính dấu phẩy động (Floating-Point Arithmetic)

3.1. Biểu diễn số nguyên

- Số nguyên không dấu (Unsigned Integer)
- Số nguyên có dấu (Signed Integer)

Số nguyên không dấu

- Biểu diễn các đại lượng luôn dương

  VD: Chiều cao, cân nặng, mã ASCII

- Tổng quát: $a_{n-1}a_{n-2} \dots a_2 a_1 a_0$
- Giá trị $\sum_{i=0}^{n-1} a_i 2^i$
- Giải biểu diễn $[0; 2^n - 1]$
- Số nguyên không dấu 1 byte
  - Bé nhất $0000\ 0000_2 = 0$
  - Lớn nhất $1111\ 1111_2 = 255 = 2^8 - 1$
- Số nguyên không dấu 1 word
  - Bé nhất $0000\ 0000\ 0000\ 0000_2 = 0$
  - Lớn nhất $1111\ 1111\ 1111\ 1111_2 = 32\ 657 = 2^{16} - 1$

Ví dụ

- Biểu diễn số nguyên không dấu sau bằng 8-bit

  A = 35, B=132

  <u>Ta có:</u>

  $A = 32+2+1=2^5+2^1+2^0$

  $A = 0010\ 0011_2$

  $B=128+4= 2^7+2^2$

  $B = 1000\ 0100_2$

- Xác định giá trị của các số nguyên không dấu biểu diễn bằng 8 bit sau
  - C = 0101 1100
  - D = 1011 1110
  - $C=2^6+2^4+2^3+2^2=64+16+8+4=92$
  - $D=2^7+2^5+2^4+2^3+2^2+2^1=128+32+16+8+4+2=190$

N=8 bit

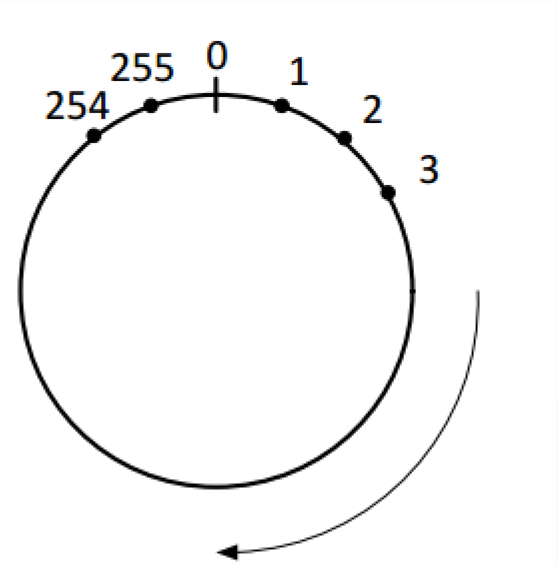
- Vấn đề đặt ra:
  - 3 + 4 = 7
  - 1 + 255 ??? ->

Biểu diễn nhị phân	Giá trị thập phân
0000 0000	0
0000 0001	1
0000 0010	2
….	
1111 1110	254
1111 1111	255

Trục số học với N = 8 bit

- Trục số học

- Trục số học máy tính



N = 16, 32, 64 bit

- n = 16 bit
  - 0000 0000 0000 0000 = 0
  - …..
  - 1111 1111 1111 1111 = 65535

  dải biểu diễn từ $0 \rightarrow 65535\ (2^{16} - 1)$:

- n = 32 bit dải biểu diễn từ $0 \rightarrow 2^{32} - 1$
- n = 64 bit dải biểu diễn từ $0 \rightarrow 2^{64} - 1$

3.1. Biểu diễn số nguyên

- **Số nguyên không dấu (Unsigned Integer)**
- **Số nguyên có dấu (Signed Integer)**

Một số cách biểu diễn

Decimal Representation	Sign-Magnitude Representation	Twos Complement Representation	Biased Representation
+8	—	—	1111
+7	0111	0111	1110
+6	0110	0110	1101
+5	0101	0101	1100
+4	0100	0100	1011
+3	0011	0011	1010
+2	0010	0010	1001
+1	0001	0001	1000
+0	0000	0000	0111
−0	1000	—	—
−1	1001	1111	0110
−2	1010	1110	0101
−3	1011	1101	0100
−4	1100	1100	0011
−5	1101	1011	0010
−6	1110	1010	0001
−7	1111	1001	0000
−8	—	1000	—

Số bù 9 & bù 10

- Số nhị phân A được biểu diễn bằng n bit:
  - Số bù 9: $A_1=(10^n-1)-A$
  - Số bù 10: $A_2= 10^n - A$
- Ví dụ: A=2101 với n=4

  A_1= 9999-2101= 7898

  A_2=10000-2101=7899

=>$A_2 = A_1 + 1$

Số bù 1 & bù 2

- Số nhị phân A được biểu diễn bằng n bit:
  - Số bù 1: $A_1=(2^n-1)-A$
  - Số bù 2: $A_2= 2^n - A$

=>$A_2 = A_1 + 1$

Ví dụ

- Với n = 8 bit cho $A = 0011\ 0100$
  - Số bù 1 của A = $(2^8 - 1) - A$

    1111 1111

    \- <u>0011 0100</u>

    1100 1011

    => đảo bit của A

- Số bù 2 của A = $2^8 - A$

  1 0000 0000

  \- <u>0011 0100</u>

  1100 1100

  => không có quy luật, thực hiện phức tạp

Quy tắc xác định số bù 1,2

- Số bù 1: Đảo các bit của số A
- Số bù 2: (Số bù 1 của A) + 1
- Ví dụ:

A	0011 0100
Số bù 1	1100 1011
	+ 1
Số bù 2	1100 1100

- Ta có:

A	0000 0000
Số bù 1	1111 1111
	+ 1
Số bù 2	1 0000 0000

(Bỏ qua bít nhớ)

=>số bù 2 của A = –A

Biểu diễn số nguyên có dấu

- Nguyên tắc: Dùng n bit

$$a_{n-1}a_{n-2}\dots a_2a_1a_0$$

$$a_{n-1}: \text{bit dấu}$$

- $A > 0$ bit $a_{n-1} = 0$, các bit còn lại biểu diễn độ lớn như số không dấu
- $A < 0$ bit $a_{n-1} = 1,$ A được biểu diễn bởi số bù hai của số dương $(-A)$

Ví dụ

- Dùng 8 bit biểu diễn số nguyên có dấu sau đây:
  - $A = +62$; $B = -76$
- Ta có:
  - $A = +62 = \mathbf{0}011\ 1110$
  - $B = -76$

    Xét số $+76 = 0100\ 1100$

    Số bù một của $= 1011\ 0011$

    $+ \qquad\qquad 1$

    Số bù hai $= 1011\ 0100$

    Vậy $B = -76 = 1011\ 0100$

    N xet: $1011\ 0100_{(2)} = 180 = 256 - 76$

Giá trị

- Dạng tổng quát của số $A > 0$ biểu diễn n bit là:

$$A = 0a_{n-2} \ldots a_2 a_1 a_0$$

- Giá trị của số dương:

$$A = \sum_{i=0}^{n-2} a_i . 2^i$$

- Dải giá trị biểu diễn cho số dương:

$$[0, (2^{n-1} - 1)]$$

Giá trị

- Dạng tổng quát của số $A < 0$ biểu diễn n bit là:

$$A = 1a_{n-2} \ldots a_2 a_1 a_0$$

- Giá trị của số âm:

$$A = -2^{n-1} + \sum_{i=0}^{n-2} a_i . 2^i$$

- Dải giá trị biểu diễn cho số âm:

$$[-1, -2^{n-1}]$$

- Vd: $B = -76 = 1011\,0100$

$=-2^7+(2^5+2^4+2^2)=-128+(32+16+4)=-128+52=-76$

Giá trị

- Dạng tổng quát của số nguyên có dấu A:

$$a_{n-1}a_{n-2} \dots a_2 a_1 a_0$$

- Giá trị của A được xác định:

$$A = -a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i$$

- Dải giá trị

$$[-(2^{n-1}), +(2^{n-1} - 1)]$$

Ví dụ

- Dùng 8 bít biểu diễn số
- Khoảng giá trị

  $[-2^7, 2^7 - 1]$ ~ [-128,127]
- Đặc điểm:
  - Trong dãy trên có 1 số 0
  - Không biểu diễn được giá trị 128

Giá trị thập phân	Biểu diễn bù 2
0	0000 0000
+1	0000 0001
+2	0000 0010
	....
+126	0111 1110
+127	0111 1111
-128	1000 0000
-127	1000 0001
...	
-2	1111 1110
-1	1111 1111

Ví dụ

- Tìm giá trị của các số nguyên có dấu biểu diễn bởi mã bù 8 bit sau:
  - $A = 0011\ 1101$
  - $B = 1010\ 0011$
- Ta có:
  - A= $2^5+2^4+2^3+2^2+2^0$

    A= 32+16+8+4+1

    A=61
  - B= $-2^7+(2^5+2^1+2^0)$

    B= -128+ (32+2+1)

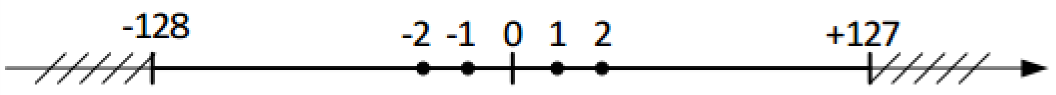
    B= -128+35

    B= - 93
  - B= $-(256-(2^7+2^5+2^1+2^0))$

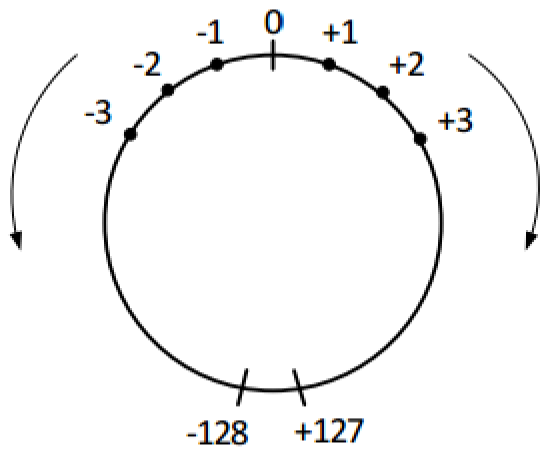
    B= - 93

TRỤC SỐ HỌC n=8 bit

- Trục số học :



- Trục số học máy tính



Ví dụ

- $n = 16$ bit: biểu diễn từ -2^{15} đến $2^{15} - 1$
  - $0000\ 0000\ 0000\ 0000 =$ 0
  - $0000\ 0000\ 0000\ 0001 =$ +1
  - ....
  - $0111\ 1111\ 1111\ 1111 =$ +32767
  - $1000\ 0000\ 0000\ 0000 =$ -32768
  - $1000\ 0000\ 0000\ 0001 =$ -32767
  - ....
  - $1111\ 1111\ 1111\ 1111 =$ -1

Mở rộng bit

- Số không dấu: Thêm các bit 0 vào bên trái
- Số có dấu
  - Số dương: thêm số 0 vào bên trái
    - $+18 = \quad\quad\quad\quad\quad 0001\ 0010$ (8 bit)
    - $+18 = 0000\ 0000\ 0001\ 0010$ (16 bit)
  - Số âm: thêm số 1 vào bên trái
    - $-18 = \quad\quad\quad\quad\quad 1110\ 1100$ (8 bit)
    - $-18 = 1111\ 1111\ 1110\ 1100$ (16 bit)

Mở rộng bit

- Chứng minh:

$$A = -2^{n-1}a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i$$

$$A = -2^{m-1}a_{m-1} + \sum_{i=0}^{m-2} 2^i a_i$$

$$-2^{m-1} + \sum_{i=0}^{m-2} 2^i a_i = -2^{n-1} + \sum_{i=0}^{n-2} 2^i a_i$$

$$2^{n-1} + \sum_{i=n-1}^{m-2} 2^i a_i = 2^{m-1}$$

$$1 + \sum_{i=0}^{n-2} 2^i + \sum_{i=n-1}^{m-2} 2^i a_i = 1 + \sum_{i=0}^{m-2} 2^i$$

$$\sum_{i=n-1}^{m-2} 2^i a_i = \sum_{i=n-1}^{m-2} 2^i$$

$$\Rightarrow \quad a_{m-2} = \cdots = a_{n-2} = a_{n-1} = 1$$

Chương 3

3.1. Biểu diễn số nguyên (Integer Representation)

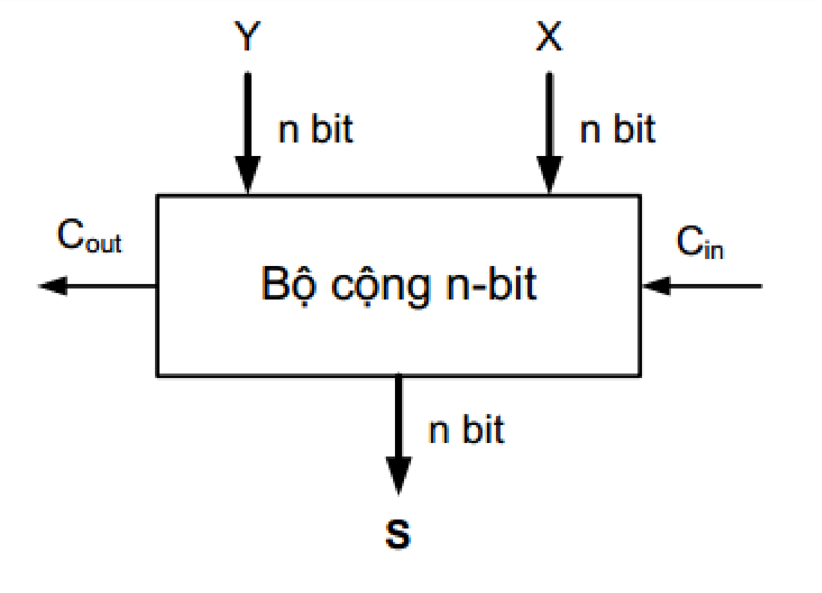
3.2. Các phép tính với số nguyên (Integer Arithmetic)

3.3. Biểu diễn số dấu phẩy động (Floating-Point Representation)

3.4. Các phép tính dấu phẩy động (Floating-Point Arithmetic)

Cộng số nguyên không dấu

- Xây dựng bộ cộng n-bit
  - $C_{out} = 0 \rightarrow$ nhận được kết quả đúng
  - $C_{out} = 1 \rightarrow$ nhận được kết quả sai



Cộng số nguyên không dấu

- Ví dụ:

57+34 = 91

Đúng/sai ?

209+73=

Đúng/sai ?

	57	=		0011 1001
+	34	=	+	0010 0010
	91			0101 1011

	209	=		1101 0001
+	73	=	+	0100 1001
	282		1	0001 1010

Cộng số nguyên có dấu

- Cộng hai số nguyên có dấu n-bit, Không quan tâm tới C_{out}
  - Hai số trái dấu: Kết quả luôn đúng
  - Hai số cùng dấu:
    - Kết quả đúng nếu cùng dấu với số hạng
    - Kết quả sai nếu trái dấu với hai số hạng
- Tràn số khi kết quả ngoài $[-(2^{n-1}),+(2^{n-1}-1)]$

Ví dụ cộng

0010 = 2 +1001 = –7 1011 = –5 (a) M = 2 = 0010 S = 7 = 0111 –S = 1001	0101 = 5 +1110 = –2 10011 = 3 (b) M = 5 = 0101 S = 2 = 0010 –S = 1110
1011 = –5 +1110 = –2 11001 = –7 (c) M = –5 = 1011 S = 2 = 0010 –S = 1110	0101 = 5 +0010 = 2 0111 = 7 (d) M = 5 = 0101 S = –2 = 1110 –S = 0010
0111 = 7 +0111 = 7 1110 = Overflow (e) M = 7 = 0111 S = –7 = 1001 –S = 0111	1010 = –6 +1100 = –4 10110 = Overflow (f) M = –6 = 1010 S = 4 = 0100 –S = 1100

Ví dụ cộng không tràn

- 70+42
- 97+(-52)
- (-90)+36
- (-74+-30)

(+ 70) = 0100 0110
+ (+ 42) = 0010 1010
+ 112 0111 0000 = +112

(+ 97) = 0110 0001
+ (- 52) = 1100 1100 (+52=0011 0100)
+ 45 1 0010 1101 = +45

(- 90) = 1010 0110 (+90=0101 1010)
+ (+36) = 0010 0100
- 54 1100 1010 = - 54

(- 74) = 1011 0110 (+74=0100 1010)
+(- 30) = 1110 0010 (+30=0001 1110)
-104 1 1001 1000 = -104

Ví dụ cộng bị tràn

- 75+82
- (-104)+(-43)

(+ 75)	=	0100 1011
+(+ 82)	=	0101 0010
+157		1001 1101

= - 128+16+8+4+1= -99 → sai

(- 104)	=	1001 1000	(+104=0110 1000)
+ (- 43)	=	1101 0101	(+ 43 =0010 1011)
- 147		1 0110 1101	

= 64+32+8+4+1= +109 → sai

Phép đảo dấu

- Bù hai số dương

+ 37	=	0010 0101		
bù một	=	1101 1010		
		+ 1		
bù hai	=	1101 1011	=	-37

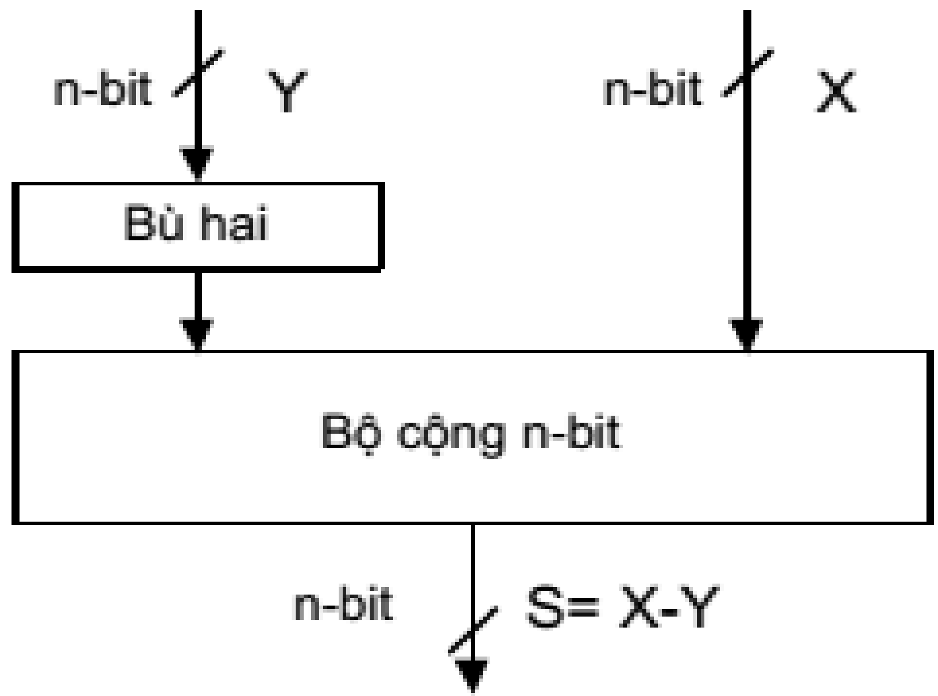
- Bù hai số âm

- 37	=	1101 1011		
bù một	=	0010 0100		
		+ 1		
bù hai	=	0010 0101	=	+37

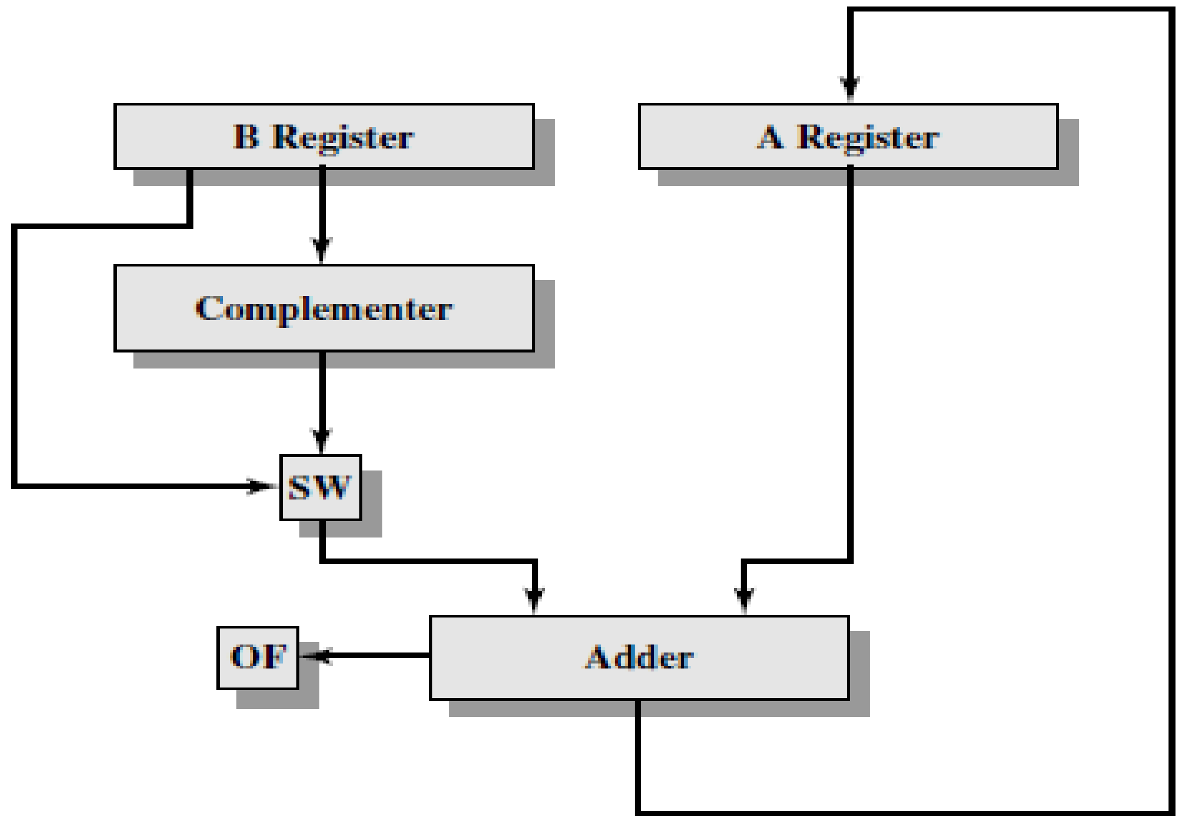
=>Phép bù 2 là phép đảo dấu số nguyên trong máy tính

Phép trừ

- $X\text{-}Y=X+(-Y)$



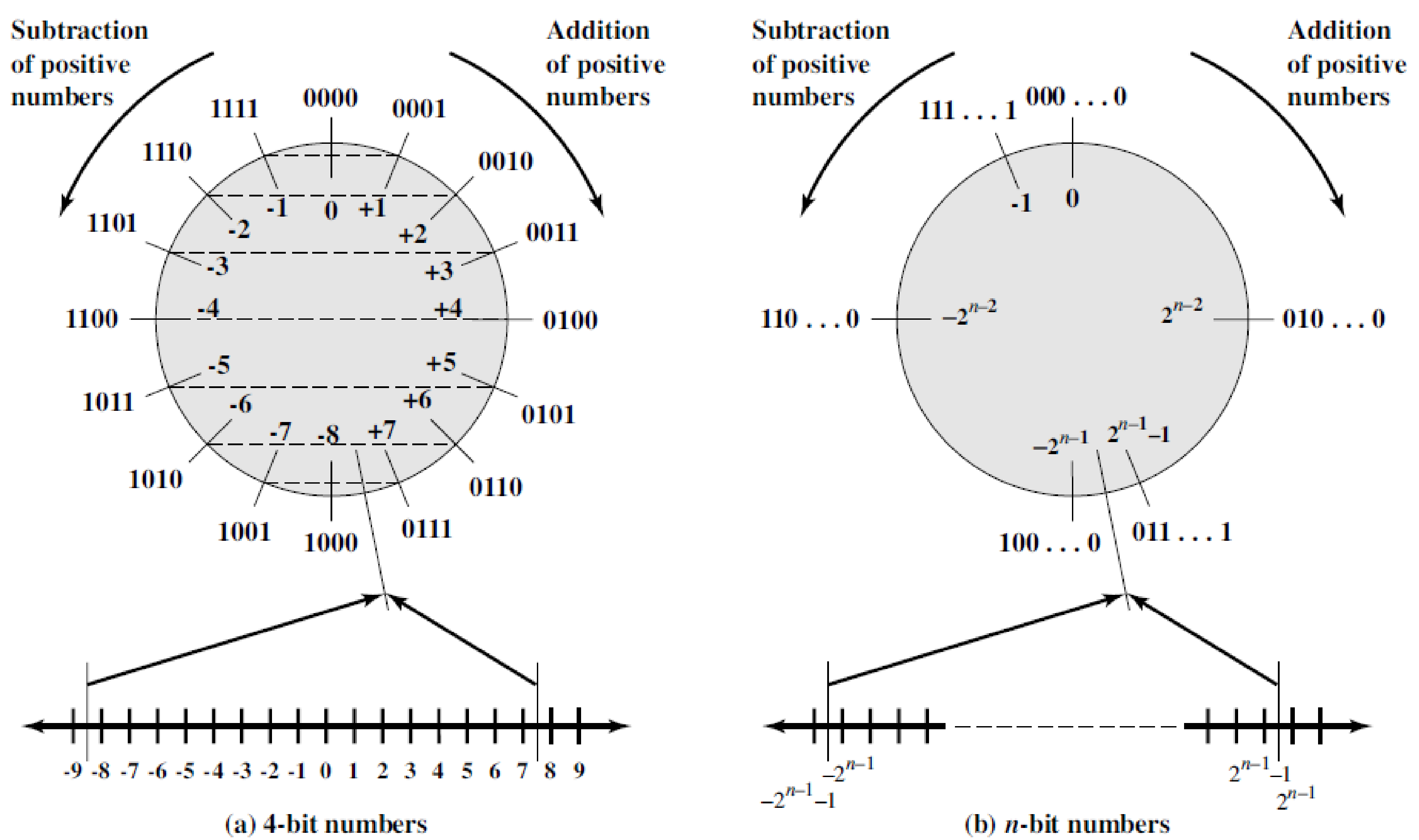
Sơ đồ bộ cộng/trừ



OF = Overflow bit
SW = Switch (select addition or subtraction)

Figure 9.6 Block Diagram of Hardware for Addition and Subtraction

Cộng/trừ nhìn trên trục số



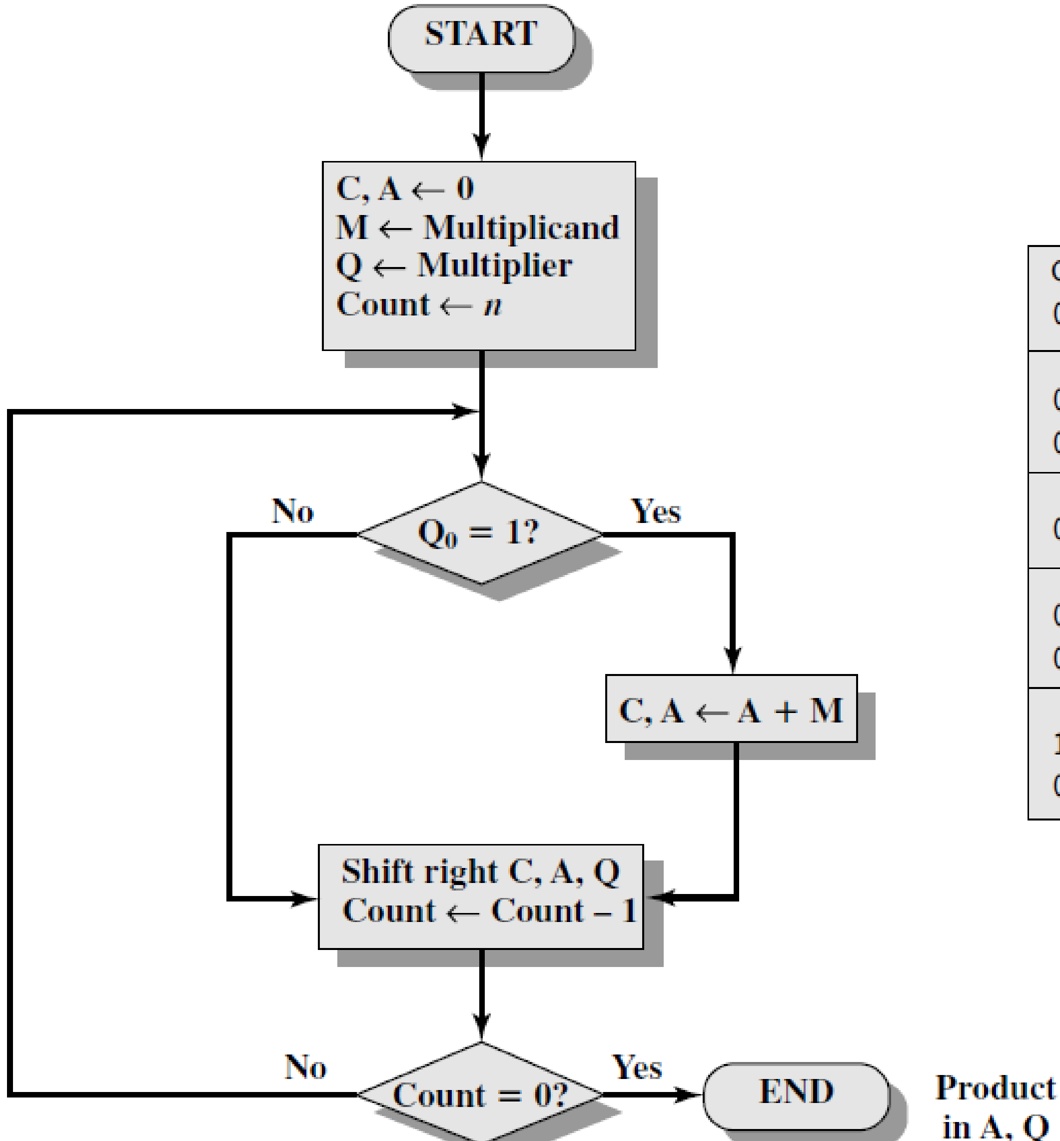
Nhân hai số không dấu

$$\begin{array}{r l}
1011 & \text{Multiplicand (11)} \\
\times 1101 & \text{Multiplier (13)} \\
\hline
1011 & \\
0000 & \\
1011 & \text{Partial products} \\
1011 & \\
\hline
10001111 & \text{Product (143)}
\end{array}$$

Nhân hai số không dấu

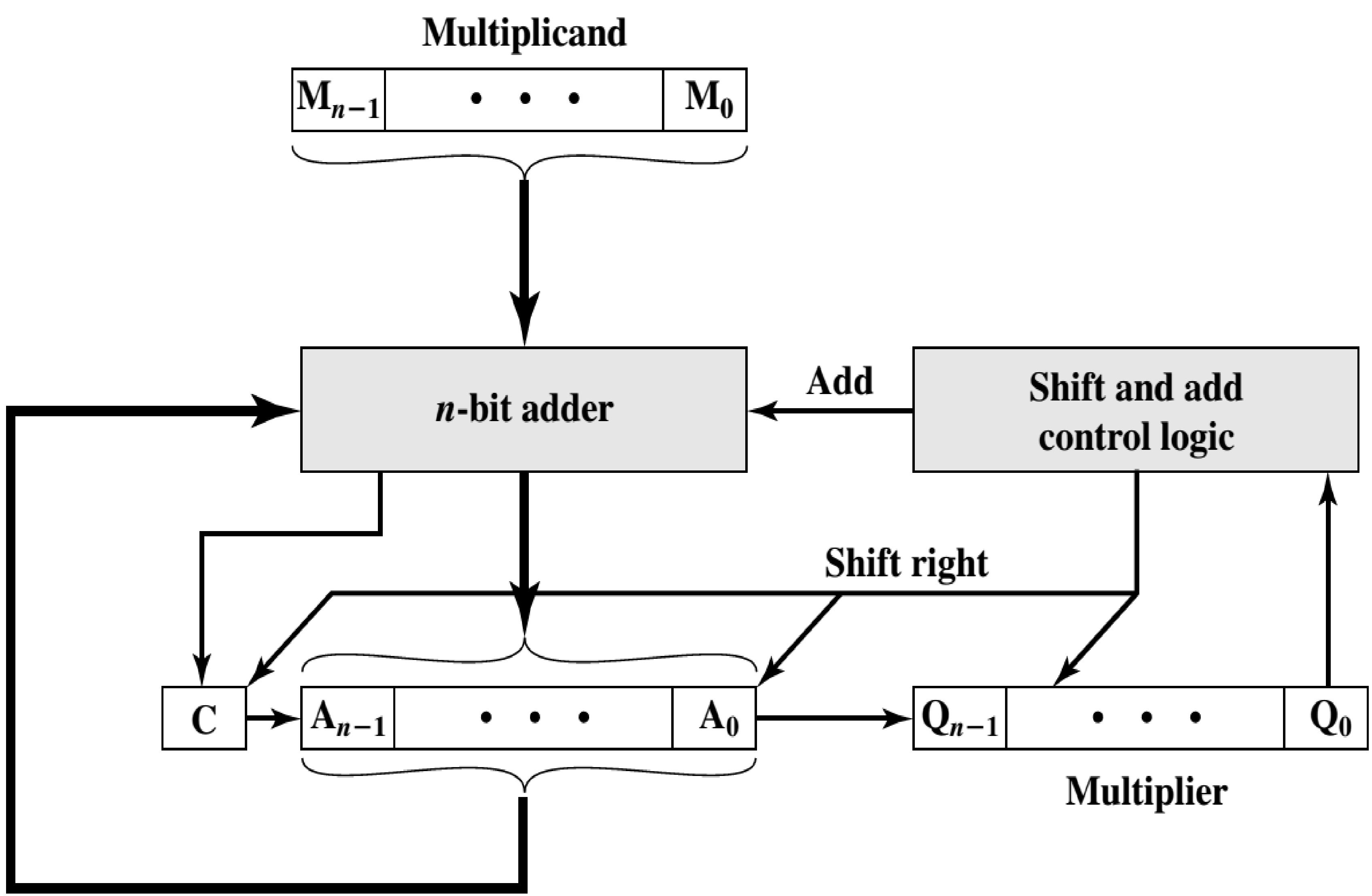
- Xác định các tích riêng phần
  - Bit của số nhân =0: tích riêng phần = 0
  - Bit của số nhân =1: tích riêng phần = số bị nhân
  - Tích riêng phần tiếp theo: dịch trái 1 bit so với tích trước đó
- Tích = tổng các tích riêng
- Chú ý: Nếu dùng 2n để biểu diễn tích của 2 số n bit thì không bị tràn

Lưu đồ bộ nhân



C	A	Q	M		
0	0000	1101	1011	Initial values	
0	1011	1101	1011	Add	First cycle
0	0101	1110	1011	Shift	
0	0010	1111	1011	Shift	Second cycle
0	1101	1111	1011	Add	Third cycle
0	0110	1111	1011	Shift	
1	0001	1111	1011	Add	Fourth cycle
0	1000	1111	1011	Shift	

Bộ nhân hai số ko dấu



Nhân hai số có dấu

- Ví dụ

1011	
$\times$ 1101	
00001011	$1011 \times 1 \times 2^0$
00000000	$1011 \times 0 \times 2^1$
00101100	$1011 \times 1 \times 2^2$
01011000	$1011 \times 1 \times 2^3$
10001111	

- Nếu ta coi là 2 số không dấu thì: 11x13=143 (đ)
- Nếu ta coi là 2 số âm: (-5)x(-3)=-113 (s)

Nhân hai số có dấu

- Sử dụng giải thuật nhân không dấu
- Sử dụng giải thuật Booth

Sử dụng giải thuật nhân không dấu

- Bước 1: Chuyển hai thừa số thành số dương tương ứng
- Bước 2: Nhân hai số dương bằng giải thuật nhân hai số không dâu
- Bước 3: Hiệu chỉnh dấu của tích
  - Nếu hai thừa cùng dấu => Giữ nguyên kết quả bbước 2
  - Nếu hai thừa số trái dấu => Đảo dấu (Lấy bù 2)

Nhân hai số có dấu

- Ví dụ

```
      0011
      0101
00000011        1011 x1 x2^0
0000000          1011 x0 x2^1
000011          1011 x1 x2^2
00000           1011 x0 x23
00001111 = 15_10
```

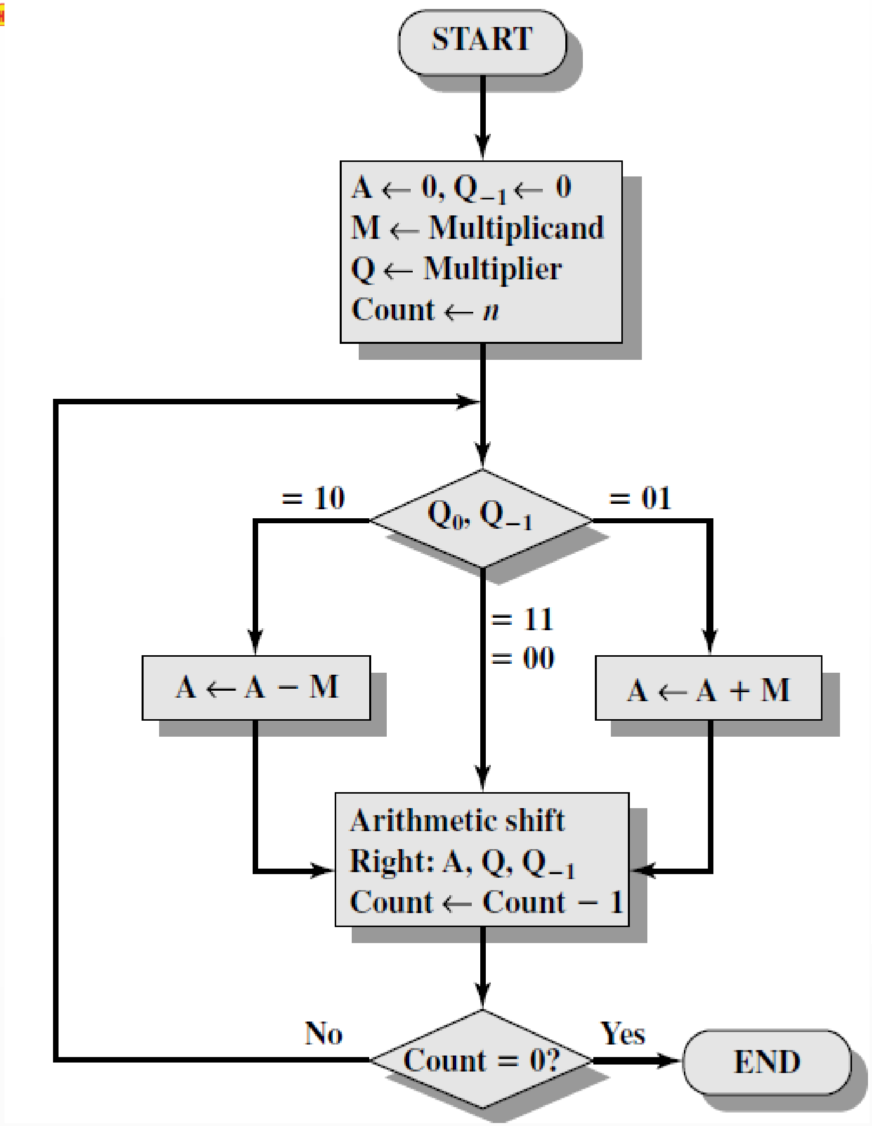
0011
$\underline{0101}$

00000011	1011 x1 x2^0
0000000	1011 x0 x2^1
000011	1011 x1 x2^2
$\underline{00000}$	1011 x0 x23
$00001111 = 15_{10}$	

Lấy bù 2: $11110001_2 = -15_{10}$

- Nếu ta coi là 2 số không dấu thì: 3x5=15 (đ)
- Nếu ta coi là 2 số âm: (-5)x(3)=(-3)x(5)=-15 (lấy bù kết quả)
- Nếu là 2 số âm (-3)x(-5)=15 (đ)

Giải thuật Booth



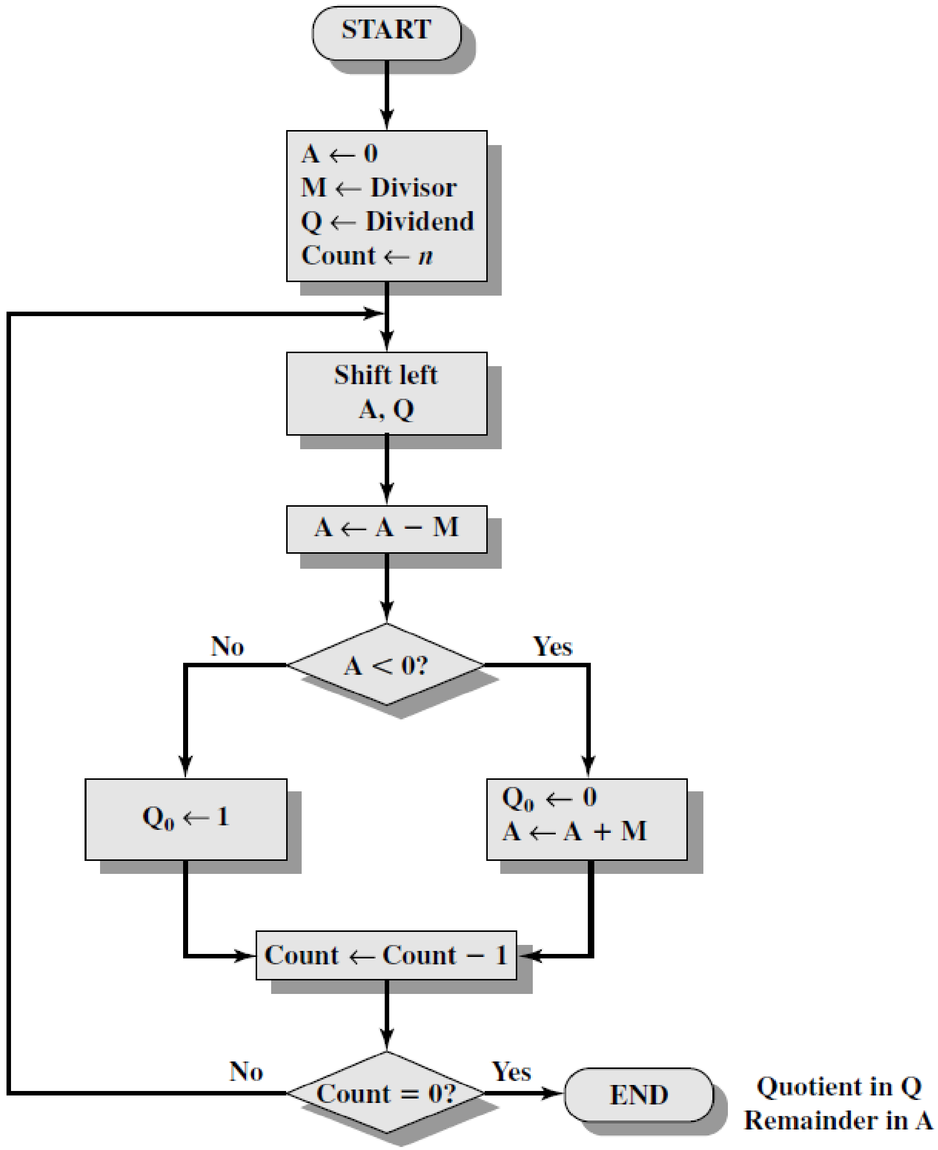
A	Q	Q_{-1}	M		
0000	0011	0	0111	Initial values	
1001	0011	0	0111	A ← A − M	First cycle
1100	1001	1	0111	Shift	
1110	0100	1	0111	Shift	Second cycle
0101	0100	1	0111	A ← A + M	Third cycle
0010	1010	0	0111	Shift	
0001	0101	0	0111	Shift	Fourth cycle

- Khi dịch sang phải thì bit A_{n-1} vừa dịch, vừa giữ nguyên để bảo toán dấu
- KQ nằm ở thanh ghi A,Q (3x7=21)

Chia hai số không dấu

Số bị chia	10010011	1011	Số chia
	- 1011	00001101	Thương
	001110		
	- 1011		
	001111		
	- 1011		
	100		Phần dư

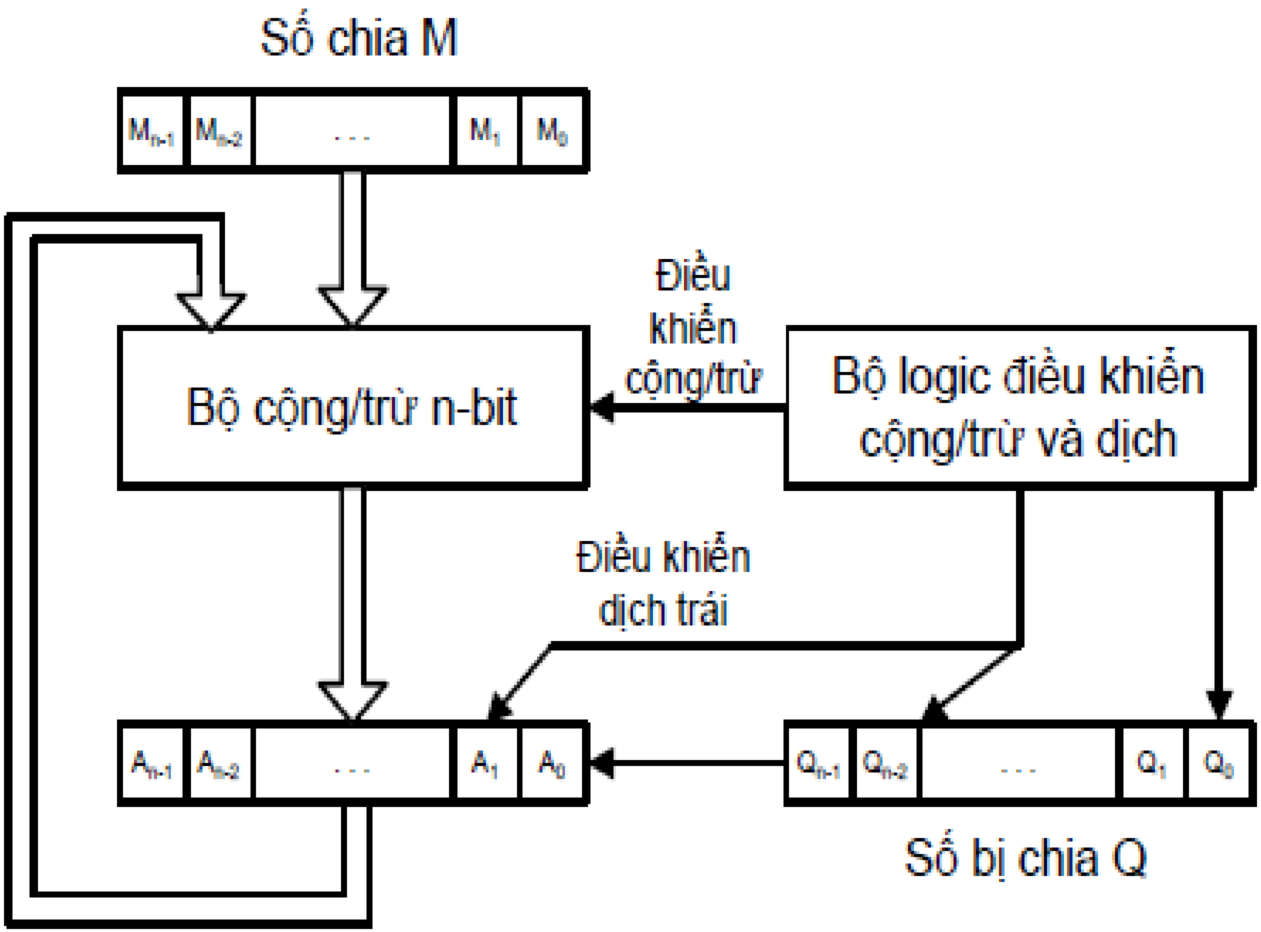
Chia hai số không dấu



A	Q	
0000	0111	Initial value
0000	1110	Shift
1101		Use twos complement of 0011 for subtraction
1101		Subtract
0000	1110	Restore, set $Q_0 = 0$
0001	1100	Shift
1101		
1110		Subtract
0001	1100	Restore, set $Q_0 = 0$
0011	1000	Shift
1101		
0000	1001	Subtract, set $Q_0 = 1$
0001	0010	Shift
1101		
1110		Subtract
0001	0010	Restore, set $Q_0 = 0$

Figure 9.17 Example of Restoring Twos Complement Division (7/3)

Bộ chia hai số không dấu



Chia số nguyên có dấu

- Bước 1: Chuyển số bị chia và số chia thành số dương tương ứng
- Bước 2: Sử dụng giải thuật chia không dấu để chia hai số dương ta được thương và số dư đều dương
- Bước 3: Hiệu chỉnh dấu của kết quả theo bảng sau

Số bị chia	**Số chia**	**Thương**	**Số dư**
dương	dương	giữ nguyên	giữ nguyên
dương	âm	đảo dấu	giữ nguyên
âm	dương	đảo dấu	đảo dấu
âm	âm	giữ nguyên	đảo dấu

Chương 3

3.1. Biểu diễn số nguyên (Integer Representation)

3.2. Các phép tính với số nguyên (Integer Arithmetic)

3.3. Biểu diễn số dấu phẩy động (Floating-Point Representation)

3.4. Các phép tính dấu phẩy động (Floating-Point Arithmetic)

Số dấu phẩy động

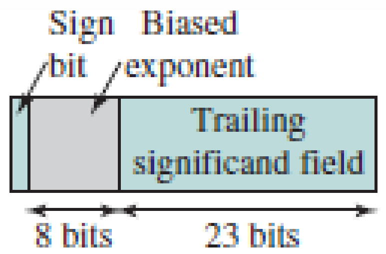
- Tổng quát:

$$X = \pm M \times R^E$$

  - M: Định trị (Mantissa)
  - R: Cơ số (Radix)
  - E: Phần mũ (Exponent)

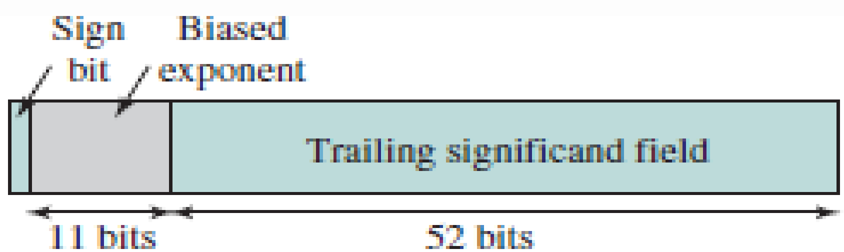
Chuẩn IEE754-2008

- Cơ số R=2
- Các dạng
  - 32 bit



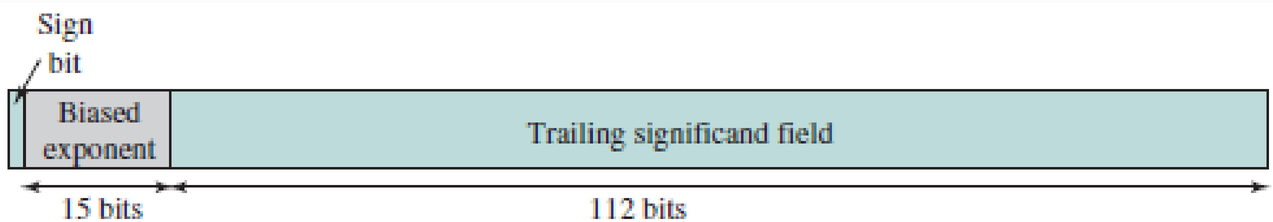
(a) Binary32 format

  - 64 bit



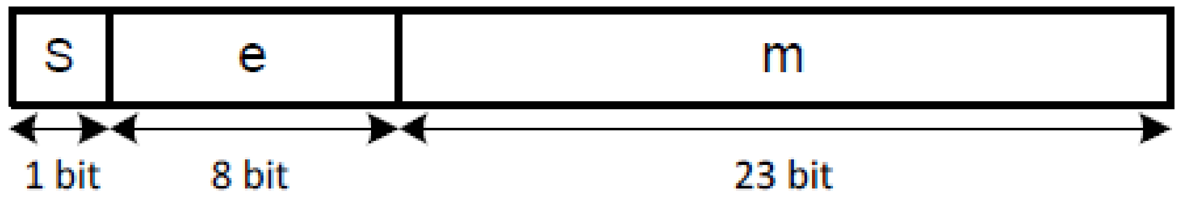
(b) Binary64 format

  - 128 bit



(c) Binary128 format

Dạng 32 bit



- S là bit dấu
  - S=0-> Số dương
  - S=1-> Số âm
- e được biểu diễn là số dịch 127 của phần mũ
  - e=E+127-> Phẫn mũ E=e-127
- m là phần lẻ của phần định trị M
  - M=1.m

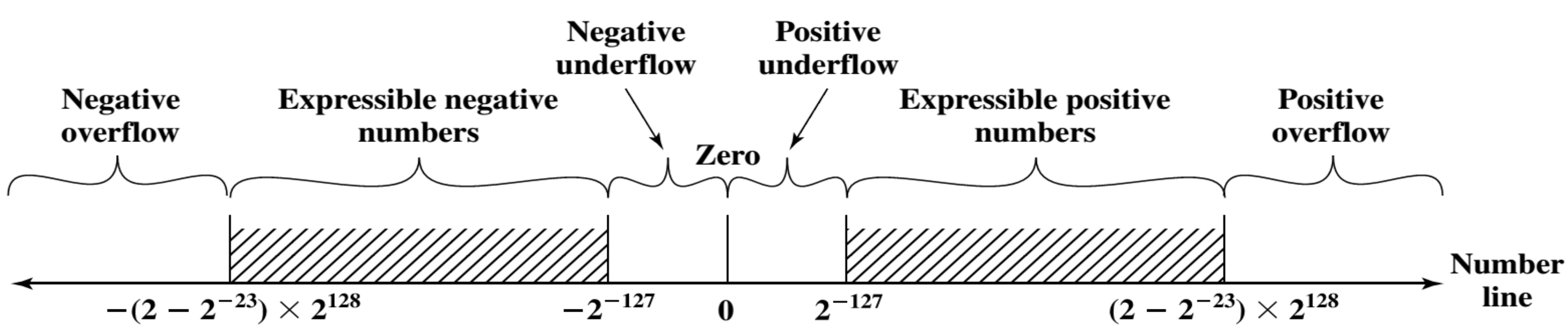
=>

$$X = (-1)^S \times 1.m \times 2^{e-127}$$

Dải giá trị dạng 32 bit

$$X = (-1)^S \times 1.m \times 2^{e-127}$$

Từ $-(2-2^{-23})\mathrm{x}2^{+128}$ đến -2^{-127}; 2^{-127} đến $(2-2^{-23})\mathrm{x}2^{+128}$



Ví dụ

- Xác định số thực sau:
  - X=1100 0001 0101 0110 0000 0000 0000 0000
    - S=1-> Số âm
    - e=1000 0010_2=130_{10}-> E=130-127=3

$$\boldsymbol{X = -1.10101100_2 \times 2^3}$$
$$\boldsymbol{= -1101.011_2 = -13.75_{10}}$$

Ví dụ

- Biễu diễn số thực 27.75

$$\boldsymbol{X = 27.7510 = 11011.11}$$
$$\boldsymbol{= 1.101111 \times 24}$$

- S=0
- e=4+127=131_{10}=10000011_2

$$\boldsymbol{X = 0100\ 0001\ 1101\ 1110\ 0000\ 0000\ 0000\ 0000}$$

Dạng 64 bit

- S là bit dấu: 1 bit
- e (11 bit) được biểu diễn là số dịch 1023 của phần mũ
  - e=E+1023-> Phẫn mũ E=e-1023
- m (52 bit) là phần lẻ của phần định trị M
  - M=1.m

=> $$X = (-1)^S \times 1.m \times 2^{e - 1023}$$

- Giải biểu diễn:

  $-(2-2^{-52})x2^{+1024}$ đến -2^{-1023}; 2^{-1023} đến $(2-2^{-52})x2^{+1023}$

Dạng 128 bit

- S là bit dấu: 1 bit
- e (15 bit) được biểu diễn là số dịch 16383 của phần mũ
  - e=E+16383-> Phẫn mũ E=e-16383
- m (112 bit) là phẩn lẻ của phần định trị M
  - M=1.m

=>

$$X = (-1)^S \times 1.m \times 2^{e\ -\ 16383}$$

- Giải biểu diễn:

-(2-2^{-112})x2^{+16384} đến -2$^{-163833}$; 2^{-16383} đến (2-2^{-112})x2^{+16384}

Các qui ước đặc biệt

- Các bit của e và m bằng 0 thì $X = \pm 0$
- Các bít của e và m bằng 1 thì $X = \pm\infty$
- Các bít của e bằng 1, m có ít nhất 1 bit bằng 1 thì nó không biểu diễn cho số nào (NAN- not a number)

Chương 3

3.1. Biểu diễn số nguyên (Integer Representation)

3.2. Các phép tính với số nguyên (Integer Arithmetic)

3.3. Biểu diễn số dấu phẩy động (Floating-Point Representation)

3.4. Các phép tính dấu phẩy động (Floating-Point Arithmetic)

Thực hiện các phép toán dấu phẩy động

- $X_1 = M_1 \times RE^1$

  $X_2 = M_2 \times RE^2$

- $X_1 \times X_2 = M_1 \times M_2 \times RE^{1+E2}$
- $X_1 / X_2 = M_1 / M_2 \times RE^{1+E2}$
- $X_1 \pm X_2 = (M_1 \times R^{E1-E2} \pm M_2) \times RE^2$
- $X_1 \times X_2 = \pm M_1 \times M_2 \times RE^{1+E2}$ $\quad E_2 \geq E_1$

Các khả năng tràn số

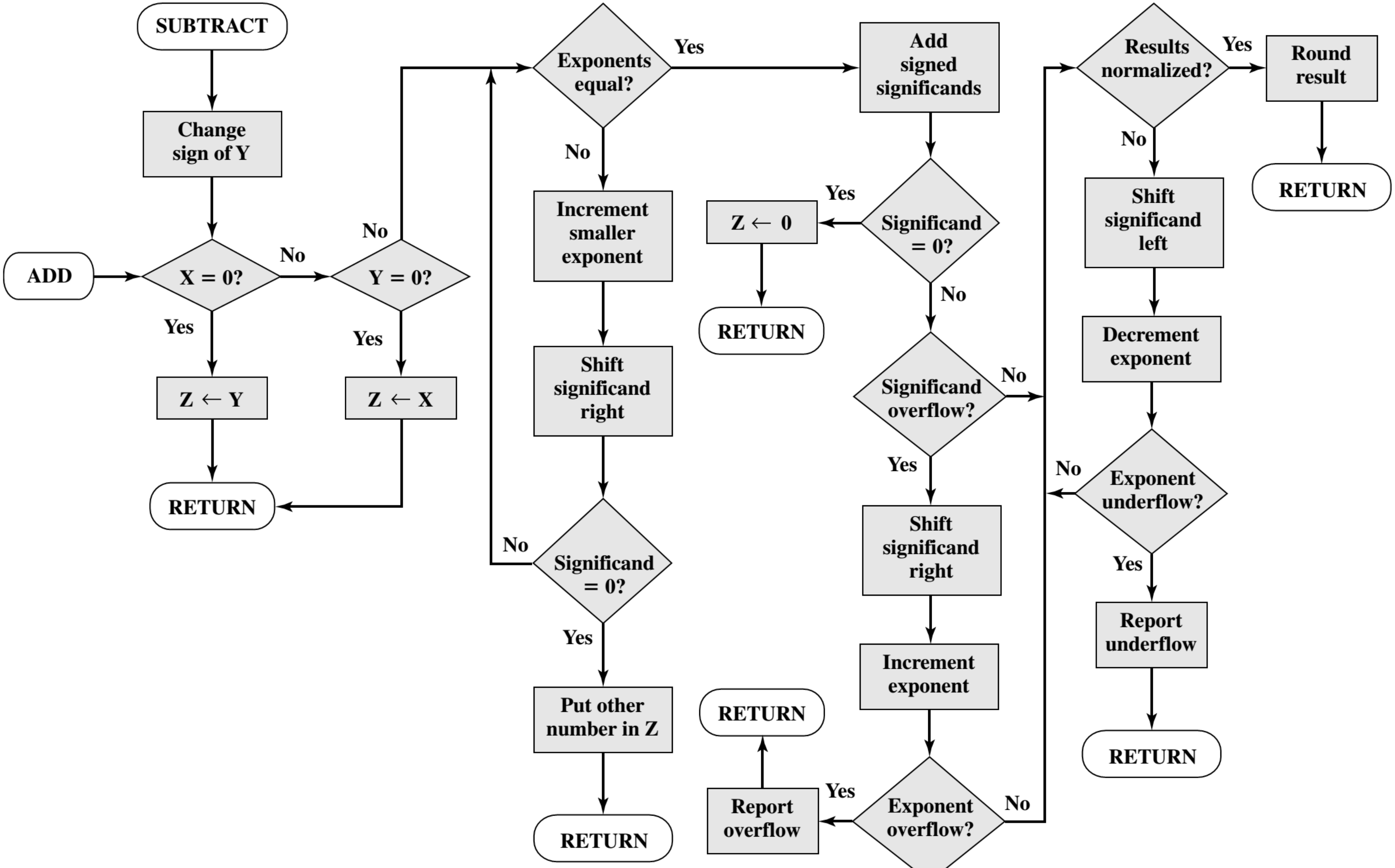
- Tràn **trên** số mũ (Exponent Overflow
- Tràn **dưới** số mũ (Exponent Underflow)
- Tràn **trên** phần định trị (Mantissa Overflown)
- Tràn **dưới** phần định trị (Mantissa Underflow)

  Các số bị mất ở bên phải phần định trị khi ***hiệu chỉnh*** phần định trị-> Làm tròn

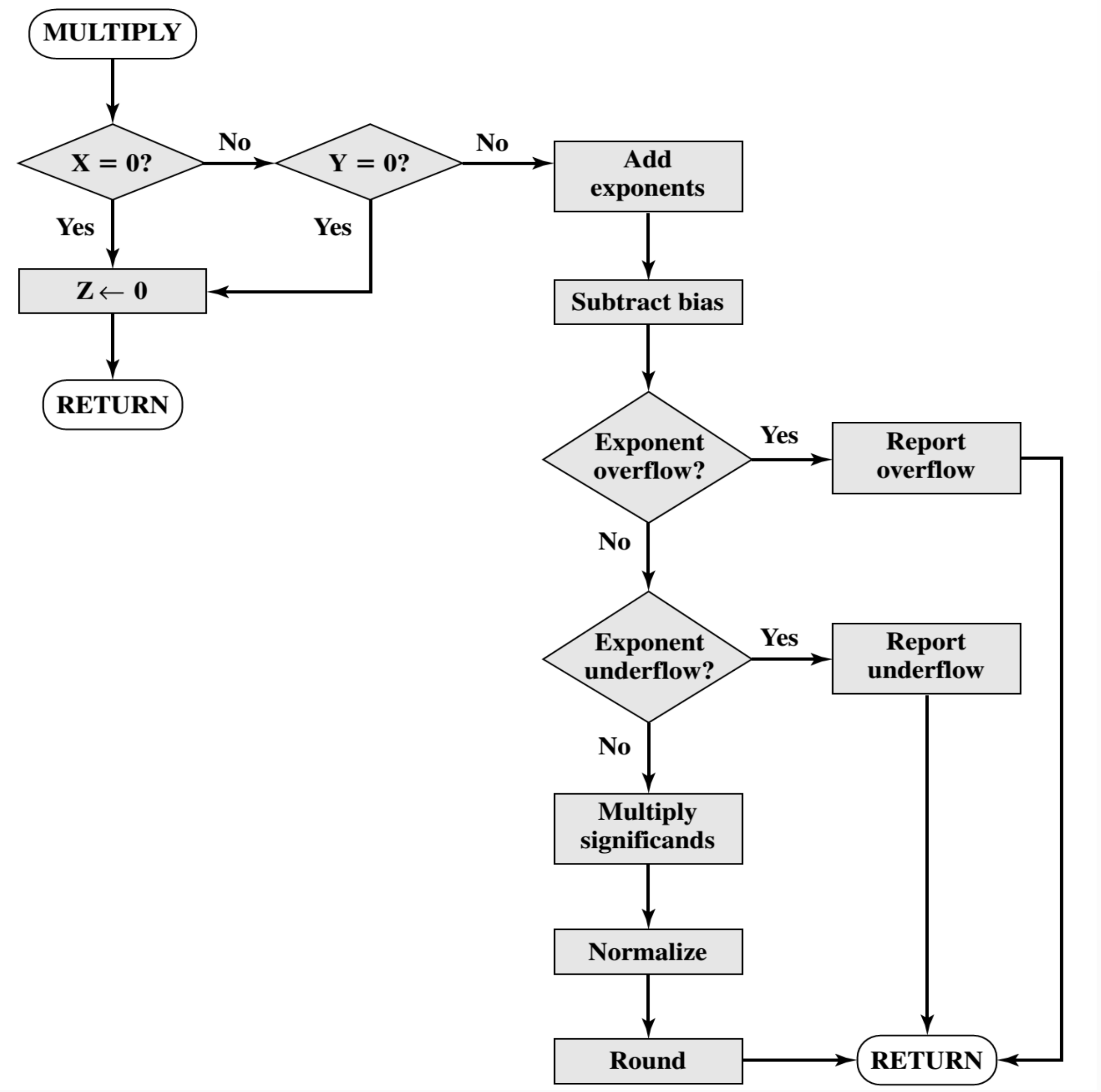
Phép cộng/ trừ dấu phẩy động

- Kiểm tra xem một trong các số hạng có bằng 0 hay không
- Qui đồng mũ, bằng cách hiệu chỉnh phần định trị
- Cộng/ trừ phần định trị
- Chuẩn hóa kết quả (Hiệu chỉnh phần định trị và mũ)

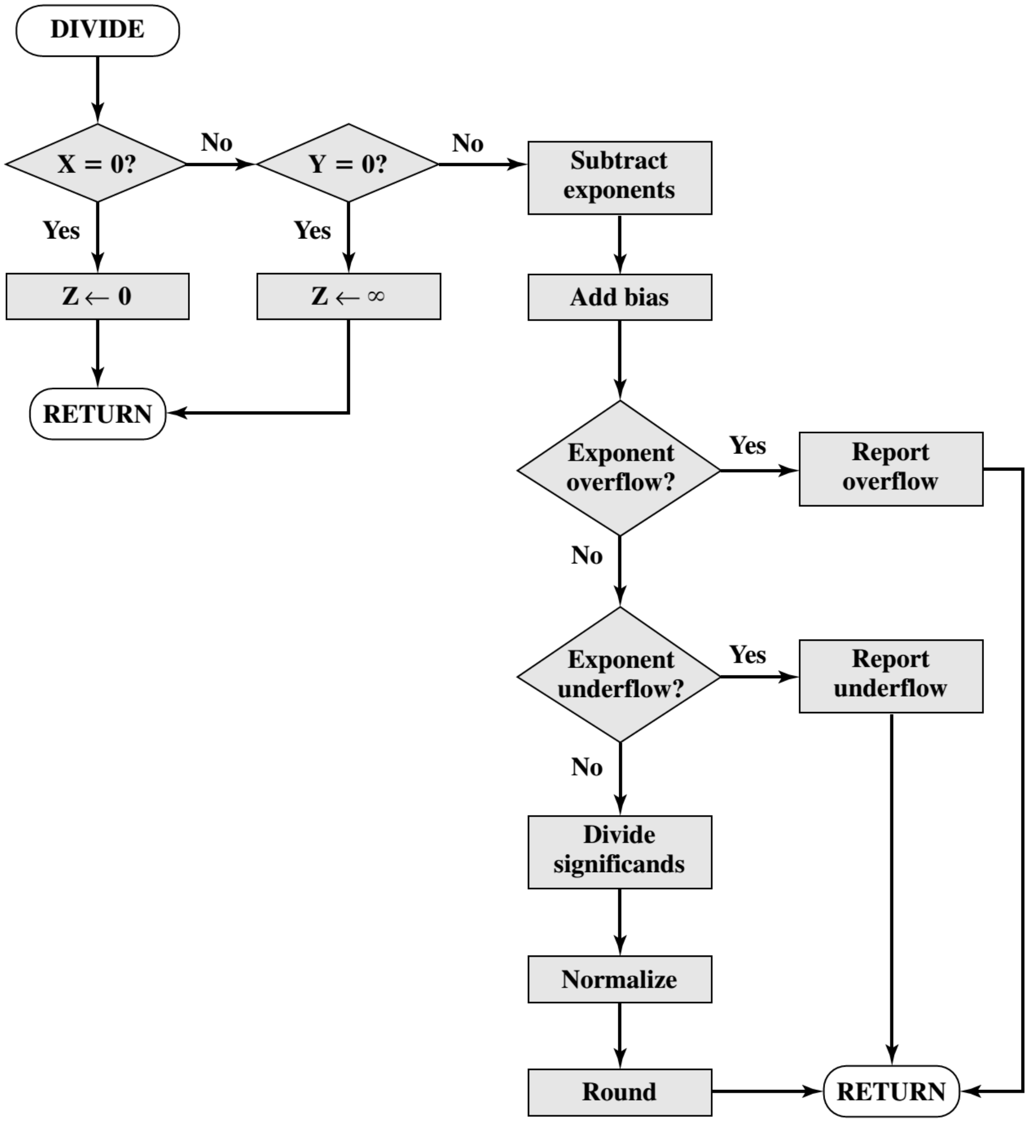
Thuật toán cộng/ trừ dấu phẩy động



Thuật toán nhân dấu phẩy động



Thuật toán chia dấu phẩy động



HẾT CHƯƠNG 3