Computer Architecture
Kiến trúc máy tính
(Lecture 3)

Luong Van Thien, PhD
Faculty of Computer Science
PHENIKAA University

(Nguồn: TS. Nguyễn Công Lượng)

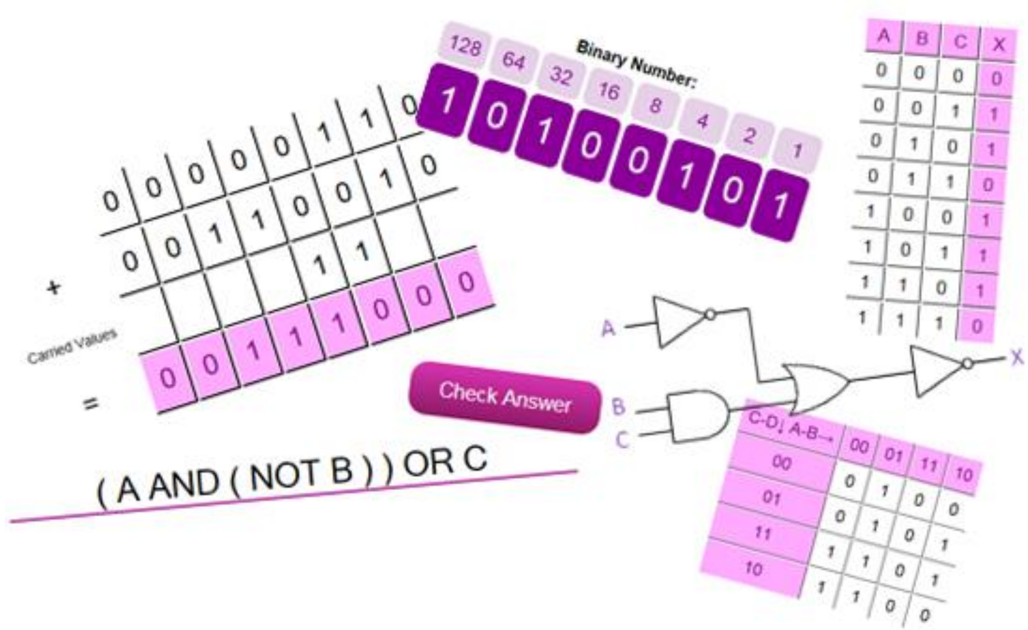
Chương 3. Số học máy tính

3.1. Biểu diễn số nguyên

3.2. Phép cộng và phép trừ số nguyên

3.3. Phép nhân và phép chia số nguyên

3.4. Số dấu phẩy động



3.1. Biểu diễn số nguyên

3.1.1. Số nguyên không dấu (Unsigned Integer)

3.1.2. Số nguyên có dấu (Signed Integer)

3.1.1. Biểu diễn số nguyên không dấu

- Nguyên tắc tổng quát: Dùng n bit biểu diễn số nguyên không dấu A:

Biểu diễn số 7 trong hệ thập phân thành số nhị phân chiều dài 3 bit

$$a_{n-1}a_{n-2}\dots a_2 a_1 a_0 \quad , a_i = 0,1$$

Giá trị của A được tính như sau:

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

Tính giá trị sau từ nhị phân sang thập phân:

10011101

11101111

10011001

Dải biểu diễn của A: từ 0 đến $2^n - 1$

3.1.1. Biểu diễn số nguyên không dấu

- n bits biểu diễn được tổng cộng 2^n số nguyên.
- Các số nguyên này có giá trị từ 0 đến $2^n - 1$
- Ví dụ: n = 2 bits, biểu diễn được tổng cộng 4 số nguyên có giá trị từ 0 đến 3
- Ví dụ: n = 8 bits, biểu diễn được tổng cộng 256 số nguyên, có giá trị từ 0 đến 255

0000 0000 = 0

0000 0001 = 1

0000 0010 = 2

0000 0011 = 3

...

1111 1111 = 255

3.1.1. Biểu diễn số nguyên không dấu

- n= 16 bit: dải biểu diễn từ 0 đến 65535 ($2^{16} - 1$)
  - 0000 0000 0000 0000 = 0
  - ...
  - 0000 0000 1111 1111 = 255
  - 0000 0001 0000 0000 = 256
  - ...
  - 1111 1111 1111 1111 = 65535

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

3.1.1. Biểu diễn số nguyên không dấu

- Ví dụ 1. Biểu diễn các số nguyên không dấu sau đây bằng 8-bit:

  $A = 41$; $B = 150$

  Giải:

  $A = 41 = 32 + 8 + 1 = 2^5 + 2^3 + 2^0$

  $41 = 0010\ 1001$

  $B = 150 = 128 + 16 + 4 + 2 = 2^7 + 2^4 + 2^2 + 2^1$

  $150 = 1001\ 0110$

3.1.1. Biểu diễn số nguyên không dấu

- Biểu diễn các số nguyên không dấu sau dưới dạng 8 bits:
- C = 70
- D = 90
- E = 135

3.1.1. Biểu diễn số nguyên không dấu

- Ví dụ 2. Cho các số nguyên không dấu M, N được biểu diễn bằng 8-bit như sau:
  - M = 0001 0010
  - N = 1011 1001

  Xác định giá trị của chúng ?

Giải:

- M = 0001 0010 = $2^4 + 2^1$ = 16 +2 = 18
- N = 1011 1001 = $2^7 + 2^5 + 2^4 + 2^3 + 2^0$

  = 128 + 32 + 16 + 8 + 1 = 185

3.1.2. Biểu diễn số nguyên có dấu

- Trong toán học, các số âm được biểu diễn bằng cách thông thường là đặt trước số dương tương ứng một dấu "−". Ví dụ: với hệ thập phân, số nguyên âm năm được biểu diễn là −5.
- Tuy nhiên, trong máy tính, khi mọi ký hiệu, con số,... đều được biểu diễn dưới hệ nhị phân thông qua hai chữ số 0 và 1 thì mọi chuyện lại trở nên phức tạp hơn.
- Có nhiều cách được sử dụng để biểu diễn số âm trong máy tính. Phần này giới thiệu cách sử dụng số bù 1, bù 2

3.1.2. Biểu diễn số nguyên có dấu

Trong hệ thập phân:

Số bù chín và Số bù mười

- Cho một số thập phân A được biểu diễn bằng n chữ số thập phân, ta có:
  - Số bù chín của A = $(10^n-1) - A$
  - Số bù mười của A = $10^n - A$
- Số bù mười của A = (Số bù chín của A) +1

3.1.2. Biểu diễn số nguyên có dấu

- Ví dụ: với n=4, cho A = 3265
  - Số bù chín của A:

$$\begin{array}{rrl} & 9999 & (10^4-1) \\ - & \underline{3265} & (A) \\ & 6734 & \end{array}$$

  - Số bù mười của A:

$$\begin{array}{rrl} & 10000 & (10^4) \\ - & \underline{3265} & (A) \\ & 6735 & \end{array}$$

3.1.2. Biểu diễn số nguyên có dấu

Trong hệ nhị phân: số bù 1 và số bù 2

- Định nghĩa: Cho một số nhị phân A được biểu diễn bằng n bit, ta có:
  - 01 Số bù một của A $= (2^n-1) - A$ 10
  - 10 Số bù hai của A $= 2^n - A = 2$ 2
- Số bù hai của A = (Số bù một của A) +1

3.1.2. Biểu diễn số nguyên có dấu

- Trước hết: nguyên tắc trừ 2 số trong hệ nhị phân:

- 0 − 0 = 0.
- 0 − 1 = −1 (viết 1 và nhớ -1)
- 1 − 0 = 1.
- 1 − 1 = 0.
- -1-1 = -10 (viết 0 nhớ -1)

- Ví dụ: tính các hiệu sau: 110 – 011, 111-101, 1001 – 0111 ?

3.1.2. Biểu diễn số nguyên có dấu

- Ví dụ: với n = 8 bit, cho A = 0010 0101
  - Số bù một của A được tính như sau:

    1111 1111 (2^8-1)

    \- <u>0010 0101</u> (A)

    1101 1010

    → đảo các bit của A
  - Số bù hai của A được tính như sau:

    1 0000 0000 (2^8)

    \- <u>0010 0101</u> (A)

    1101 1011

    → thực hiện khó khăn

Tìm số bù 2 của:

A= 1001

B = 1010 1100

C= 0101 0110

D= 0101 1010

3.1.2. Biểu diễn số nguyên có dấu

- Số bù một của A = đảo giá trị các bit của A
- (Số bù hai của A) = (Số bù một của A) + 1
- Ví dụ:
  - Cho A = 0010 0101
  - Số bù một = 1101 1010

    \+ 1
  - Số bù hai = 1101 1011
- Nhận xét:

  A = 0010 0101

  Số bù hai = + 1101 1011

  1 0000 0000 = 0

  (bỏ qua bit nhớ ra ngoài)

  → Số bù hai của A = -A

3.1.2. Biểu diễn số nguyên có dấu

Nguyên tắc tổng quát: Dùng n bit biểu diễn số nguyên có dấu A:

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

- Với A là số dương: bit $a_{n-1} = 0$, các bit còn lại biểu diễn độ lớn như số không dấu
- Với A là số âm: được biểu diễn bằng số bù hai của số dương tương ứng, vì vậy bit $a_{n-1} = 1$

3.1.2. Biểu diễn số nguyên có dấu

Cách biểu diễn số dương:

- Dạng tổng quát của số dương A:

$$0a_{n-2}\ldots a_2a_1a_0$$

- Giá trị của số dương A: (ở hệ thập phân)

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

- Dải biểu diễn cho số dương: 0 đến $2^{n-1}-1$

3.1.2. Biểu diễn số nguyên có dấu

Cách biểu diễn số âm:

- Dạng tổng quát của số âm A:

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

- Giá trị của số âm A: (ở hệ thập phân)

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

- Dải biểu diễn cho số âm: -1 đến -2^{n-1}

3.1.2. Biểu diễn số nguyên có dấu

Cách biểu diễn một số nguyên có dấu tổng quát:

- Dạng tổng quát của số nguyên A:

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

Cần thống nhất trước đây là số nguyên có dấu

- Giá trị của A được xác định như sau:

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

Ví dụ: tính giá trị của số nhị phân có dấu:

A=1010

B=1011

C=0101

- Dải biểu diễn: từ $-(2^{n-1})$ đến $+(2^{n-1}-1)$

Ví dụ: dải biểu diễn của số nguyên có dấu với n = 3 bit, 4 bit ?

3.1.2. Biểu diễn số nguyên có dấu

- Ví dụ 2. Hãy xác định giá trị của các số nguyên có dấu được biểu diễn dưới đây:
  - P = 0110 0010
  - Q = 1101 1011

  Giải:
  - P = 0110 0010 = $-0*2^7 + 64+32+2 = +98$
  - Q = 1101 1011 = -128+64+16+8+2+1 = -37

3.1.2. Biểu diễn số nguyên có dấu

- Ví dụ 1. Biểu diễn các số nguyên có dấu sau đây bằng 8-bit:

  A = +58 ; B = -80

Giải:

A = +58 = 32+ 16 + 8 + 2

= $2^5 + 2^4 + 2^3$

0011 1010

B = -80

Ta có: + 80 = 0101 0000

Số bù một = 1010 1111

\+ 1

Số bù hai = 1011 0000

Số âm: được biểu diễn bằng số bù hai của số dương tương ứng

3.1.2. Biểu diễn số nguyên có dấu

- Ví dụ 1 (tiếp) : Biểu diễn các số nguyên có dấu dưới đây bằng 8 bits:

C= - 40

D = -50

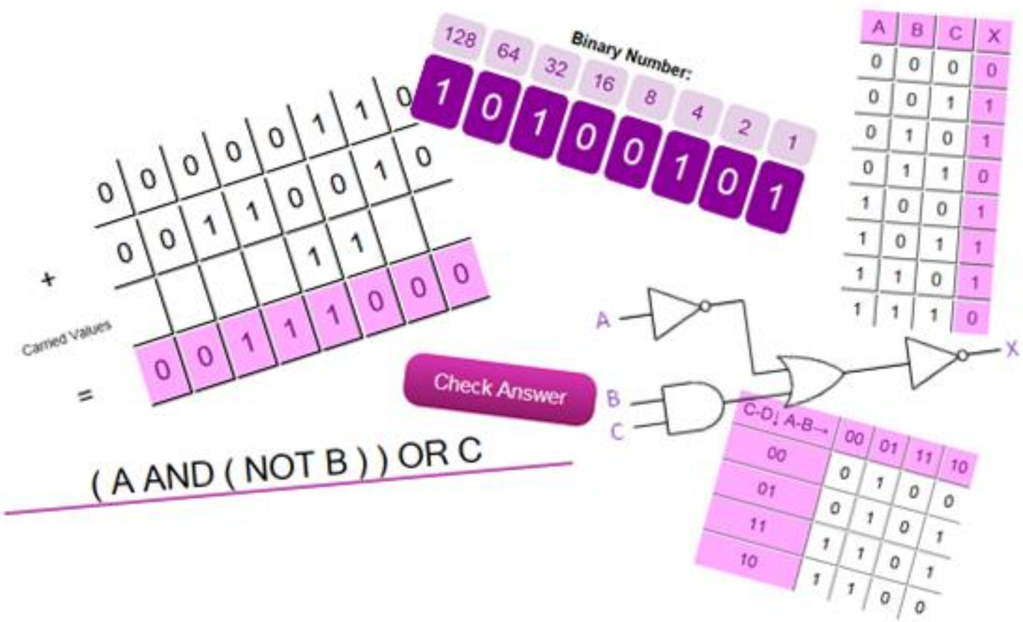
Chương 3. Số học máy tính

3.1. Biểu diễn số nguyên

3.2. Phép cộng và phép trừ số nguyên

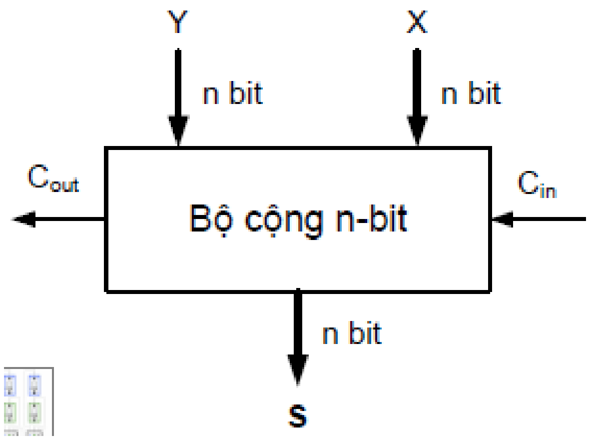
3.3. Phép nhân và phép chia số nguyên

3.4. Số dấu phẩy động



3.2.1. Phép cộng số nguyên

- Phép cộng số nguyên không dấu:



Khi cộng hai số nguyên không dấu n-bit, kết quả nhận được là n-bit:

- Nếu C_{out}=0 → nhận được kết quả đúng.
- Nếu C_{out}=1 → nhận được kết quả sai, do tràn nhớ ra ngoài *(Carry Out)*.
- Tràn nhớ ra ngoài khi: tổng > $(2^n - 1)$

3.2.1. Phép cộng số nguyên

- Phép cộng số nguyên không dấu: Ví dụ

	57	=		0011 1001
+	34	=	+	0010 0010
	91			0101 1011 = 64+16+8+2+1=91 → đúng

	209	=		1101 0001
+	73	=	+	0100 1001
	282		1	0001 1010
				0001 1010 = 16+8+2=26 → sai

→ có tràn nhớ ra ngoài (C_{out}=1)

Để có kết quả đúng ta thực hiện cộng theo 16-bit:

	209 =		0000 0000 1101 0001
+	73 =	+	0000 0000 0100 1001
			0000 0001 0001 1010 = 256+16+8+2 = 282

3.2.1. Phép cộng số nguyên

- Phép cộng số nguyên có dấu: nguyên tắc sau:

  Khi cộng hai số nguyên có dấu n-bit, kết quả nhận được là n-bit và không cần quan tâm đến bit C_{out}.

- Cộng hai số khác dấu: kết quả luôn luôn đúng.
- Cộng hai số cùng dấu:
  - nếu dấu kết quả cùng dấu với các số hạng thì kết quả là đúng.
  - nếu kết quả có dấu ngược lại, khi đó có tràn xảy ra (Overflow) và kết quả bị sai.
- Tràn xảy ra khi tổng nằm ngoài dải biểu diễn:

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

3.2.1. Phép cộng số nguyên

- Phép cộng số nguyên có dấu: ví dụ: Cộng các số nguyên có dấu (biểu diễn 8 bits) sau:
- A = + 70 + (+42)
- B = (+97) + (-52)
- C = (-90) + (+36)
- D = (-74) + (-30)

3.2.1. Phép cộng số nguyên

- Phép cộng số nguyên có dấu: ví dụ:
  - (+ 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

3.2.1. Phép cộng số nguyên

- Phép cộng số nguyên có dấu: ví dụ:

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

= - 128+16+8+4+1= -99 → sai

Tràn số vì tổng vượt quá dải biểu diễn từ [-128,+127]

3.2.2. Phép trừ số nguyên

- Phép trừ hai số nguyên: X-Y = X+(-Y)
- Nguyên tắc: Lấy bù hai của Y để được –Y, rồi cộng với X

Ví dụ: thực hiện các phép trừ sau:

A= 34-22

B= 50 – 18

C= 80 - 50

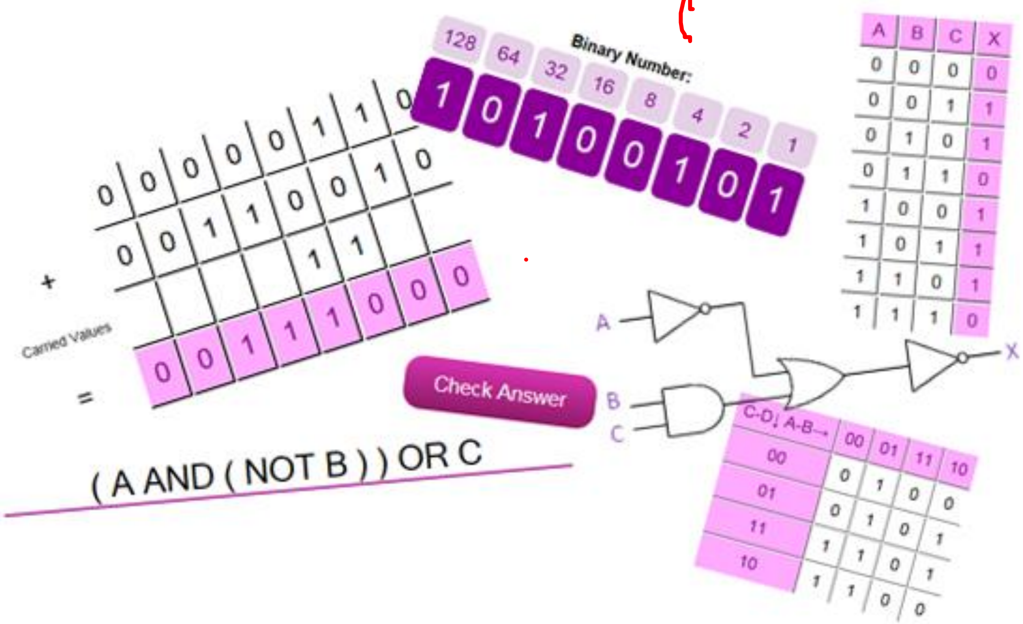
Chương 3. Số học máy tính

3.1. Biểu diễn số nguyên

3.2. Phép cộng và phép trừ số nguyên

3.3. Phép nhân và phép chia số nguyên VHDL

3.4. Số dấu phẩy động



3.3.1. Phép nhân

- Nhân số nguyên

1011	Số bị nhân	(11)
x 1101	Số nhân	(13)
1011		
0000	Các tích riêng phần	
1011		
1011		
10001111	Tích	(143)

3.3.1. Phép nhân

- Nguyên tắc nhân số nguyên:
  - Các tích riêng phần được xác định như sau:
    - Nếu bit của số nhân bằng 0 → tích riêng phần bằng 0.
    - Nếu bit của số nhân bằng 1 → tích riêng phần bằng số bị nhân.
    - Tích riêng phần tiếp theo được dịch trái một bit so với tích riêng phần trước đó.
  - Tích bằng tổng các tích riêng phần
  - Nhân hai số nguyên n-bit, tích có độ dài 2n bit (không bao giờ tràn).

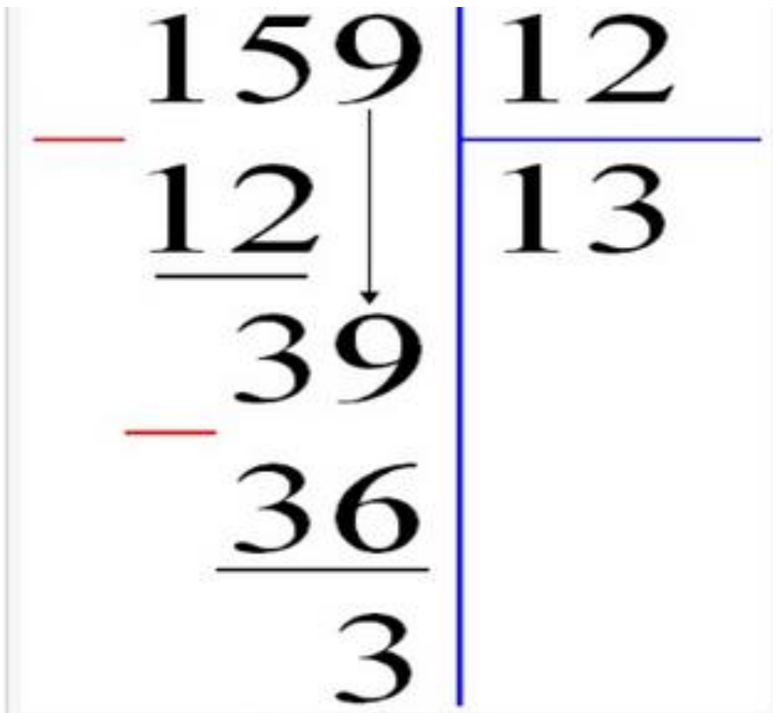
3.3.1. Phép nhân

- Nhân số nguyên không dấu A và B sau:
- A = 0010, B = 0011

- A = 0110, B = 0101

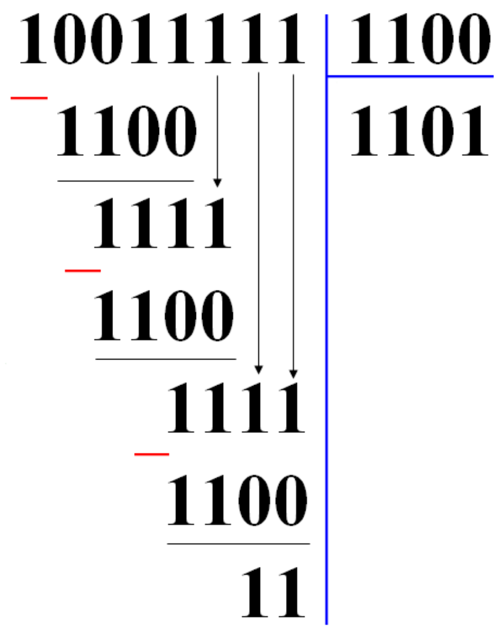
3.3.2. Phép chia

- Chia 2 số nhị phân giống với chia 2 số thập phân



3.1.4. Phép chia

- Ví dụ 1: Chia số 10011111 cho 1100:



- Ví dụ 2: Chia số 64 cho 8
- Chia số 154 cho 12

3.4. Biểu diễn số dấu phẩy động

1. Nguyên tắc chung

- Floating Point Number → biểu diễn cho số thực
- Tổng quát: một số thực X được biểu diễn theo kiểu số dấu phẩy động như sau:

  $$X = M * R^E$$

  - M là phần định trị (Mantissa),
  - R là cơ số (Radix),
  - E là phần mũ (Exponent).

$$976{,}000{,}000{,}000{,}000 = 9.76 \times 10^{14}$$

$$0.0000000000000976 = 9.76 \times 10^{-14}$$

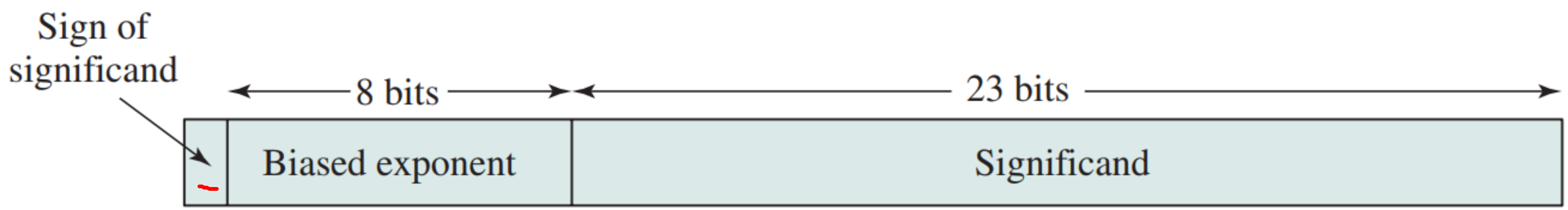
0.11×2^5

110×2^2

0.0110×2^6

1.1×2^4 (Dạng tiêu chuẩn)

3.4. Ví dụ



(a) Format

$1.1010001 \times 2^{10100}$	=	0	10010011	10100010000000000000000	=	1.6328125×2^{20}
$-1.1010001 \times 2^{10100}$	=	1	10010011	10100010000000000000000	=	-1.6328125×2^{20}
$1.1010001 \times 2^{-10100}$	=	0	01101011	10100010000000000000000	=	1.6328125×2^{-20}
$-1.1010001 \times 2^{-10100}$	=	1	01101011	10100010000000000000000	=	$-1.6328125 \times 2^{-20}$

(b) Examples

Figure 10.18 Typical 32-Bit Floating-Point Format

Hệ thập lục phân

Hệ đếm thập lục phân (Hexadecimal)

- **FEDC,76$_{16}$=**
  - **15•4096+14•256+13•16+12•1+7•1/16+6•1/256**
  - **15•16^3+14•16^2+13•16^1+12•16^0+7•16^{-1}+6•16^{-2}**
  - **r = cơ số (r = 16), d=digit (0 ≤ d ≤ F), m = số chữ số trước dấu phẩy, n = số chữ số sau dấu phẩy**

$$H = \sum_{i=-n}^{m-1} d_i \cdot 16^i$$

Ví dụ: 1011 0111B = ?H

Color	Colorname	Hex	(R,G,B)
	Black	#000000	(0,0,0)
	White	#FFFFFF	(255,255,255)
	Red	#FF0000	(255,0,0)
	Lime	#00FF00	(0,255,0)
	Blue	#0000FF	(0,0,255)
	Yellow	#FFFF00	(255,255,0)
	Cyan	#00FFFF	(0,255,255)
	Magenta	#FF00FF	(255,0,255)
	Silver	#C0C0C0	(192,192,192)
	Gray	#808080	(128,128,128)
	Maroon	#800000	(128,0,0)
	Olive	#808000	(128,128,0)
	Green	#008000	(0,128,0)
	Purple	#800080	(128,0,128)
	Teal	#008080	(0,128,128)
	Navy	#000080	(0,0,128)

Chuẩn IEEE754

- Định dạng số thực theo IEEE754

S	E	m

- **S** : là bit dấu của phần định trị M:
  - S = 0 : số dương
  - S = 1 : số âm
- **e** : là mã thừa n (excess-n) của phần mũ E, n là số bit biểu diễn số của E (do đó không cần lưu bit dấu cho E)
  - $e = E + (2^n-1)$ → $E = e - (2^n-1)$
- **m** : là phần lẻ của phần định trị M ở dạng chuẩn:
  - M = 1.m (Chú ý: Không sử dụng số bù 2)
- Công thức xác định giá trị của số thực:
  - $X = (-1)^S \times 1.m \times 2^E$

Ví dụ

- Ví dụ 1: Xác định giá trị của số thực được biểu diễn bằng 32-bit $X=C1560000_{(16)}$:
  - $X=1\underline{100\ 0001\ 0}101\ 0110\ 0000\ 0000\ 0000\ 0000_{(2)}$
  - $S = 1 \rightarrow$ số âm
  - $e = 1000\ 0010_{(2)} = 130_{(10)} \rightarrow E = 130-127=3$
- Vậy
  - $X = -1.10101100 * 2^3 = -1101.011_{(2)} = -13.375_{(10)}$

Ví dụ

– Ví dụ 2: Biểu diễn số thực X= 83.75 về dạng số dấu chấm động IEEE754 dạng 32-bit

Giải:

- $X = 83.75_{(10)} = 1010011.11_{(2)} = 1.01001111 \times 2^6$

Ta có:

- S = 0 vì đây là số dương
- E = e-127 = 6 →
- $e = 127 + 6 = 133_{(10)} = 1000\ 0101_{(2)}$

Vậy:

- $X = 0\underline{100\ 0010\ 1}010\ 0111\ 1000\ 0000\ 0000\ 0000_{(2)}$
- $X = 42A78000_{(16)}$

Mở rộng phạm vi

$$+18 = 00010010 \quad \text{(Dấu – độ lớn, 8 bit)}$$

$$+18 = 0000000000010010 \quad \text{(Dấu – độ lớn, 16 bit)}$$

$$-18 = 10010010 \quad \text{(Dấu – độ lớn, 8 bit)}$$

$$-18 = 1000000000010010 \quad \text{(Dấu – độ lớn, 16 bit)}$$

Với dạng bù hai, quy tắc trên không còn đúng, ta có ví dụ sau

$$+18 = 00010010 \quad \text{(Bù hai, 8 bit)}$$

$$+18 = 0000000000010010 \quad \text{(Bù hai, 16 bit)}$$

$$-18 = 11101110 \quad \text{(Bù hai, 8 bit)}$$

$$-32658 = 1000000011101110 \quad \text{(Bù hai, 16 bit)}$$

Vậy, với dạng bù hai, để mở rộng phạm vi, ta thêm vào bên trái của số đó các bit giống với bit dấu. Quy tắc này ta có thể gọi là quy tắc mở rộng dấu.

$$-18 = 11101110 \quad \text{(Bù hai, 8 bit)}$$

$$-18 = 1111111111101110 \quad \text{(Bù hai, 16 bit)}$$