

计算机组成理论第三期

-计算机中的数值表示 (2)-

大连理工大学立命馆大学国际信息软件学部大森孝之

讲座内容

- ➡ 二元运算符
- 扩展移位运
- 算加法/减法
- 初步内容
-

二进制符号扩展

- 如何用更多位表示相同的值
 - 例如, 8 位二进制 \rightarrow 16 位二进制
- 新值的高位是原值的最高位
填好就OK
 - 负数以 2 的补码表示为前提

8bit 00001111 15_{10}

16bit 000000000000001111

8bit 11110001 -15_{10}

16bit 111111111111110001

符号扩展法的合法性

■ 如果新值的高位用原值的最高位填充就可以了。

■ 不言自明，如果是肯定的

■ 如果为负

8bit 11110001 -15₁₀

16bit 1111111111110001

-2¹⁵ 2¹⁴ 2⁸ 2⁷

$$-2^n = 2^n - 2^{n+1}$$

$$-2^n = 2^n + 2^{n+1} - 2^{n+2}$$

⋮

$$-2^7 = 2^7 + 2^8 + \dots + 2^{14} - 2^{15}$$

确认问题

- (1) 将 001010102 符号扩展为 16 位。
- (2) 将 101010102 符号扩展为 16 位。
- (3) 通过确认 (2) 中转换前后的十进制数匹配，确保执行的符号扩展正确。



翻译狗
www.fanyigou.com
www.translateGo.com

讲座内容

- 二元运算符
- 扩展移位运
- 算加法/减
- ➡ 法
- 初步内容

换挡操作

■ 向左或向右移动二进制位的操作

■ 对于 1 位左移

$$00001100_2 = 8 + 4 = 12_{10}$$



↓ 2次

标志是暂时的
不要想

$$00011000_2 = 16 + 8 = 24_{10}$$

丢弃最重要的位

勒斯比特是 0

■ 对于 2 位右移

$$00011000_2 = 16 + 8 = 24_{10}$$



↓ 1/4次

$$00000110_2 = 4 + 2 = 6_{10}$$

换档操作

■ 注意左右边缘

左移 1
位

$$11000000_2 = 128 + 64 = 192_{10}$$



$$10000000_2 = 128 = 128_{10}$$

不加倍

右移 1
位

$$00001001_2 = 8 + 1 = 9_{10}$$



$$00000100_2 = 4 = 4_{10}$$

1/2次
(四舍五入到最近的整数)

算术移位

■ 如果签了

■ 算术移位

右移 1 位

$$11111000_2 = -8_{10}$$



$$11111100_2 = -4_{10}$$

左移 1 位

$$11111000_2 = -8_{10}$$



$$11110000_2 = -16_{10}$$

逻辑转移

■ 未签名
■ 逻辑转移

如果已签名，则
不要进行逻辑转
换

向右 1 位
转移

$11111000_2 = 248_{10}$ $\times -8_{10}$
 $\searrow \searrow \searrow \searrow \searrow \searrow \searrow \searrow$
 $01111100_2 = 124_{10}$ 124_{10}

左移 1 位

$01111000_2 = 120_{10}$ 120_{10}
 $\swarrow \swarrow \swarrow \swarrow \swarrow \swarrow \swarrow \swarrow$
 $11110000_2 = 240_{10}$ $\times -16_{10}$

班次计算注意事项

■ 注意标志的有无

(如果有符号则算术移位)

■ 不总是 $\times 2^n$, $/ 2^n$

■ 溢出舍入

■ 00000011 右侧的 1 位算术移位 (310)

* 00000001 (1_{10})

11111101 右侧的 1 位算术移位 ($= -310$)

* 11111110 (-2_{10})

确认问题

- (1) 算术左移 111010102 1 位。
- (2) 算术将 101010102 向右移动 1 位。
- (3) 将 101010102 逻辑左移 1 位。
- (4) 将 101010102 逻辑右移 1 位。



翻译狗
www.fanyigou.com
www.translateGo.com

讲座内容

- 二元运算符
- 扩展移位运
- 算加法/减
- 法
- ➡ 初步内容

二元加法

■ 正数+正数

$$\begin{array}{r} 000001111 \\ + 000000010 \\ \hline \end{array}$$

00010001

15_{10}

2_{10}

正确的

17_{10}

带十进制数
你可以用同样的
方式做长除法

$$\begin{array}{r} 01111110 \\ + 00000010 \\ \hline \end{array}$$

10000000

126_{10}

2_{10}

溢出

128_{10}

(溢出: 溢出)

如何处理它取决于编
程语言等。

✖

这是-128

二进制减法

■ 只做正数+负数

$$\begin{array}{r} 00001111 \quad 15_{10} \\ + 11111110 \quad -2_{10} \\ \hline 00001101 \quad 13_{10} \end{array}$$

正确的

忽略这里的进位

$$\begin{array}{r} 00001111 \quad 15_{10} \\ + 11101101 \quad -19_{10} \\ \hline \end{array}$$

正确的

二进制减法

负数+负数

$$\begin{array}{r} 11110001 \\ + 11111110 \\ \hline 11101111 \end{array}$$

-15_{10}

-2_{10}

-17_{10}

正确的

$$\begin{array}{r} 10000011 \\ + 11111100 \\ \hline 01111111 \end{array}$$

-125_{10}

-4_{10}

-129_{10}

溢出



这是 127

另外溢出条件

- 对于正数和正数，负数和负数
 - 计算结果符号与原值不同则溢出
- 对于正数和负数
 - 不溢出

确认问题

- 有两个二进制数 A 和 B 。
求 $A-B$ 结果时
首先，对要减去的数 B 取 (1)
并求与 A 的和。



翻译狗
www.fanyigou.com
www.translateGo.com

确认问题

■ 以 8 位二进制计算以下内容

■ (1) $(-1) + 1$

■ (2) $3 + (-2)$

■ (3) $3 + (-5)$

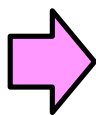
■ 用二进制表示要相加的两个数，然后将它们相加。



翻译狗
www.fanyigou.com
www.translateGo.com

讲座内容

- 二元运算符
- 扩展移位运
- 算加法/减
- 法
- 初步内容



二进制乘法

■ 正数 × 正数

- 可以像十进制数一样计算的位数是乘数的位数与被乘数的位数之和。
-

$$\begin{array}{r} 0011 \\ \times 0110 \\ \hline 0000 \\ 0011 \\ 0011 \\ 0000 \\ \hline 00010010 \end{array}$$

3_{10}
 6_{10}
 18_{10}

二进制乘法

■ 正数 x 负数

- 对乘数和被乘数进行符号扩展并截断长除法的高位
-

$$\begin{array}{r} 00000011 \\ \times 1111010 \\ \hline \end{array}$$

3_{10}
 -6_{10}

$$\begin{array}{r} 00000011 \\ 00000011 \\ 00000011 \\ 00000011 \\ 00000011 \\ 00000011 \\ 00000011 \\ \hline \end{array}$$

截短

$$11101110 \quad -18_{10}$$

二进制乘法

■ 负数 x 负数

■ 与正 x 负相同

■ 它可以通过将其转换为正数 x 正数来计算。

$$\begin{array}{r} 11111101 \quad -3_{10} \\ \times 11111010 \quad -6_{10} \\ \hline 11111101 \\ 11111101 \\ 11111101 \\ 11111101 \\ 11111101 \\ 11111101 \\ 11111101 \\ \hline 11111101 \quad 18_{10} \\ \hline \text{截短} \quad 00010010 \end{array}$$

二分法

■ 正数／正数

■ 示例) $1310/510 = 2$ 余数 $3 = 11012/01012$

$$\begin{array}{r} 10 \\ 101 \overline{) 1101} \\ \underline{101} \\ 11 \end{array}$$

■ 可以像十进制数一样计算

二分法

■ 签名分区

■ 除法 = 商 x 除数 + 余数
■ 数示例)

■ $(+7) / (+2) = (+3) \text{ 余数 } (+1)$

■ $(-7) / (-2) = (+3) \text{ 余数 } (-1)$

■ $(-7) / (+2) = (-3) \text{ 余数 } (-1)$

■ $(+7) / (-2) = (-3) \text{ 余数 } (+1)$

✖ $(+7) / (-2) = (-4) \text{ 余数 } (-1)$

被除数的符号和余数的符号
必须相同

二进制十进

- 第一个小数位，使用像十进制数一样的小数点，相当于 2^{-1} 。

- 示例) $11.11_2 = 1 * 2^1 + 1 * 2^0 + 1 * 2^{-1} + 1 * 2^{-2}$
 $= 2 + 1 + 0.5 + 0.25$
 $= 3.75_{10}$

$$\begin{aligned} 0.0011_2 &= 1 * 2^{-3} + 1 * 2^{-4} \\ &= 0.125 + 0.0625 \\ &= 0.1875_{10} \end{aligned}$$

二进制十进制问题

■ 发生错误

■ 例) $0.110 = 0.000110011001100 \dots_2$
 $= 0.0625 + 0.03125 + \dots$

十进制有限小数不能用有限
数字表示!

关于算术左移

左移 1 位

$10101010 = -86_{10}$

$\downarrow \diagup \diagup \diagup \diagup \diagup \diagup$

$11010100 = -44_{10}$

溢出

- 负数的 1 位算术左移
如果无符号位的最左边位为 0，则溢出
- 如果为 1，则不会溢出
* 实际上，即使是逻辑左移，移位结果也是一样的（溢出时的结果不同）

负 $11xxxxxx$ 正 $00xxxxxx$

$\diagdown \diagdown \diagdown \diagdown \diagdown \diagdown \diagdown$ $\diagdown \diagdown \diagdown \diagdown \diagdown \diagdown \diagdown$

$1xxxxxx0$ $0xxxxxx0$

即使在算术移位
逻辑移位相同
* 单词算术左移
有些想法没用

参考

- Computer Configuration and Design 5th Edition by David A. Patterson, John L. Hennessy, 成田光明翻译, Nikkei BP