



中国科学院大学
University of Chinese Academy of Sciences

B0911006Y-01

2024-2025学年春季学期

计算机组成原理

Principles of Computer Organization

第 7 讲 计算机中数的运算I

如何进行定点移位计算？

主讲教师：石 侃

shikan@ict.ac.cn

2025年3月17日

6.3 定点运算

一、移位运算

1. 移位的意义

$$15.\text{m} = 1500.\text{cm}$$

小数点右移 2 位

机器用语 15 相对于小数点 左移 2 位
(小数点不动)

左移 绝对值扩大

右移 绝对值缩小

在计算机中，移位与加减配合，能够实现乘除运算

2. 算术移位规则(对有符号数的移位)

6.3

不论正负，移位后的符号位不变！

真值	码 制	除符号位外 添补代码
正数	原码、补码、反码	添 0
负数	原 码	添 0
	补 码	左移 添 0
		右移 添 1
	反 码	添 1

添加的0
有意义
还是1有
意义？

①原码除符号位外“各位取反、末位加1”形成补码

②原码除符号位外“各位取反”形成反码

③因此，以负数补码数值位中最后的1为分界，右边(含1)同原码，左边同反码

例6.16

6.3

设机器数字长为 8 位（含 1 位符号位），写出 $A = +26$ 时，三种机器数左、右移一位和两位后的表示形式及对应的真值，并分析结果的正确性。

解： $A = +26 = +11010$ 则 $[A]_{\text{原}} = [A]_{\text{补}} = [A]_{\text{反}} = 0,0011010$

移位操作	机 器 数	对应的真值
	$[A]_{\text{原}} = [A]_{\text{补}} = [A]_{\text{反}}$	
移位前	0,0011010	+26
左移一位	0,0110100	+52
左移两位	0,1101000	+104
右移一位	0,0001101	+13
右移两位	0,0000110	+6



例6.16

6.3

设机器数字长为8位（含1位符号位），写出 $A = +26$ 时，三种机器数左、右移一位和两位后的表示形式及对应的真值，并分析结果的正确性。

解： $A = +26 = +11010$ 则 $[A]_{\text{原}} = [A]_{\text{补}} = [A]_{\text{反}} = 0,0011010$

移位操作	机 器 数	对应的真值
	$[A]_{\text{原}} = [A]_{\text{补}} = [A]_{\text{反}}$	
移位前	0,0011010	+26
左移一位	0,0110100	+52
左移两位	0,1101000	+104
右移一位	0,0001101	+13
右移两位	0,0000110	+6

右移最低
位丢1，
影响精度

若继续左移，即左移三位，则最高位丢1，结果为+80，出错

例6.17

6.3

设机器数字长为 8 位（含 1 位符号位），写出 $A = -26$ 时，三种机器数左、右移一位和两位后的表示形式及对应的真值，并分析结果的正确性。

解： $A = -26 = -11010$

原码

移位操作	机 器 数	对应的真值
移位前	1,0011010	-26
左移一位		
左移两位		
右移一位		
右移两位		



例6.17

6.3

设机器数字长为 8 位（含 1 位符号位），写出 $A = -26$ 时，三种机器数左、右移一位和两位后的表示形式及对应的真值，并分析结果的正确性。

解： $A = -26 = -11010$

原码	移位操作	机 器 数	对应的真值
	移位前	1,0011010	-26
	左移一位	1,0110100	-52
	左移两位	1,1101000	-104
	右移一位		
	右移两位		

例6.17

6.3

设机器数字长为 8 位（含 1 位符号位），写出 $A = -26$ 时，三种机器数左、右移一位和两位后的表示形式及对应的真值，并分析结果的正确性。

解： $A = -26 = -11010$

原码

移位操作	机 器 数	对应的真值
移位前	1,0011010	-26
左移一位	1,0110100	-52
左移两位	1,1101000	-104
右移一位	1,0001101	-13
右移两位	1,0000110	-6

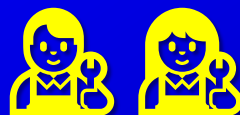
补码

移位操作	机 器 数	对应的真值
移位前	1,1100110	-26
左移一位	1,1001100	-52
左移两位	1,0011000	-104
右移一位	1,1110011	-13
右移两位	1,1111001	-7

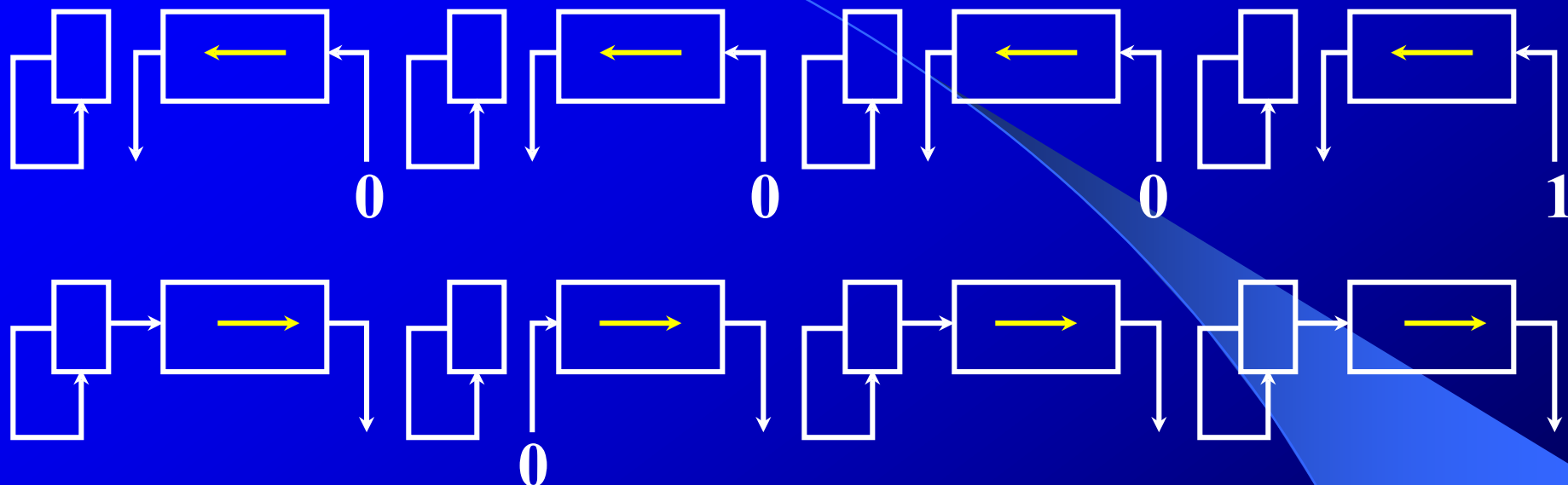
反码

移位操作	机 器 数	对应的真值
移位前	1,1100101	-26
左移一位	1,1001011	-52
左移两位	1,0010111	-104
右移一位	1,1110010	-13
右移两位	1,1111001	-6

3. 算术移位的硬件实现



6.3



(a) 真值为正

(b) 负数的原码

(c) 负数的补码

(d) 负数的反码

← 丢 1 出错

← 丢 1 出错

← 丢 0 出错

← 丢 0 出错

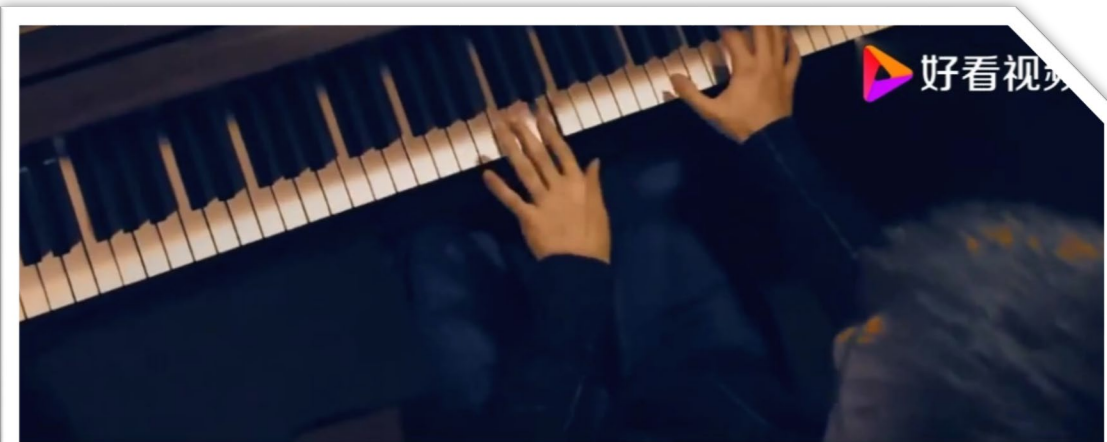
→ 丢 1 影响精度

→ 丢 1 影响精度

→ 丢 1 影响精度

→ 丢 0 影响精度

上述情况下丢1或0导致出错或精度损失，是因为这个1或0代表实际数值
若把上述1或0改变成0或1，则结果是：不会出错，或不会影响精度



<https://www.idm314.org/>

<https://www.piday.org/>

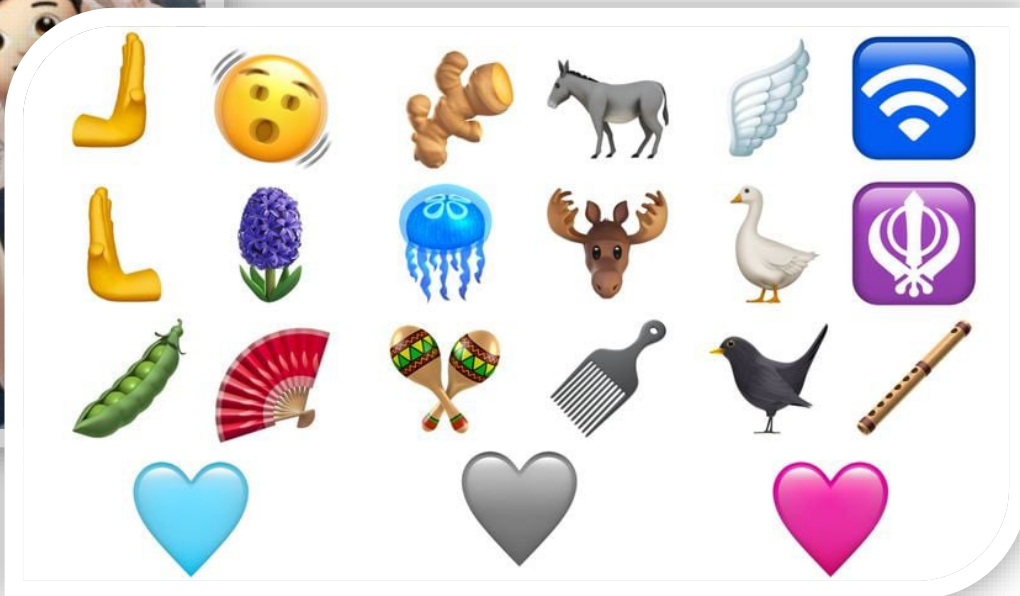
<https://mathigon.org/>



The History of Unicode



据说，在标题中添加表情符号能提升关注度



<https://blog.emojipedia.org/first-look-new-emojis-in-ios-16-4/>
<https://www.macrumors.com/2023/03/08/apple-releases-ios-16-4-public-beta-3/>

The Most Frequently Used **Emoji** of 2021

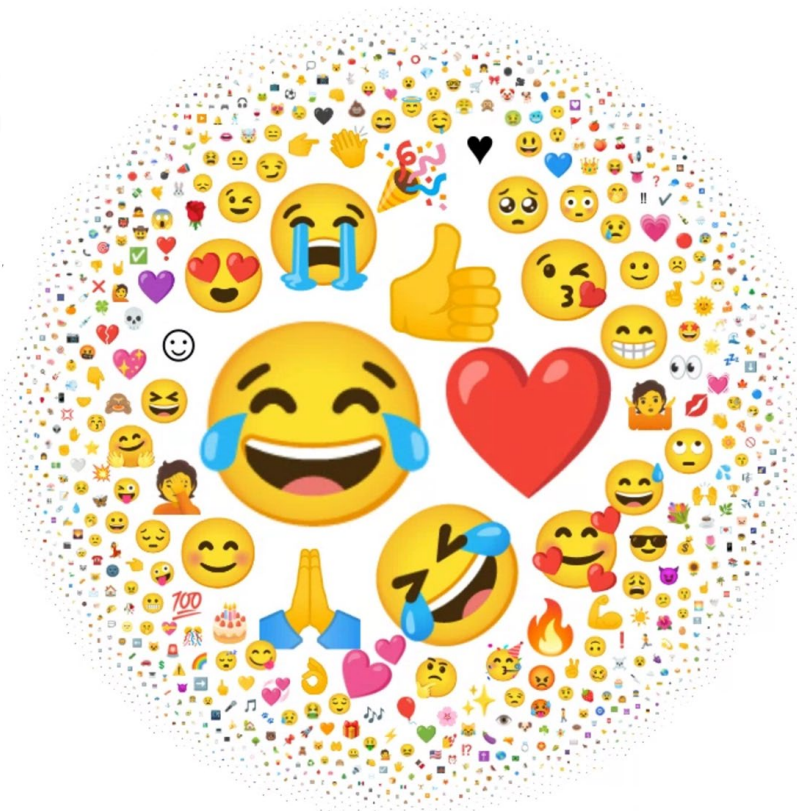
U+1F602

Unicode Code Points
统一码编码点

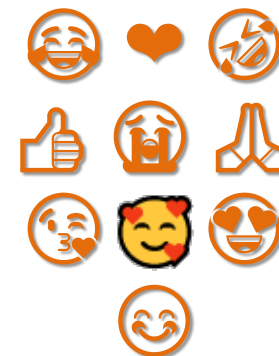


Face with
Tears of Joy

喜极而泣
破涕为笑
“笑cry了”



The top ten
emoji used
worldwide:



<https://home.unicode.org/emoji/emoji-frequency/>
<http://unicode.org/Public/emoji/14.0/emoji-test.txt>
<https://emojipedia.org/emoji/%F0%9F%98%82/>
<https://www.cnblogs.com/redcat8850/p/15717881.html>
<https://www.jianshu.com/p/d46d0fdbad93>
<https://zhuanlan.zhihu.com/p/41203455>

☺ 复习:

1. 浮点数的表示

|数符|阶码(指数): 定点整数|尾数: 定点小数| 【基不用表示】

2. 浮点数的范围

① 与阶码位数和基大小有关

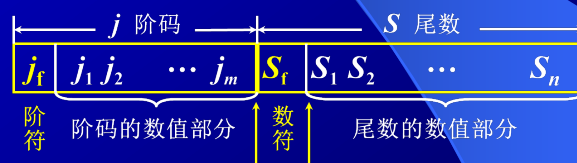
② 正/负上溢(中断溢出)、正/负下溢(按机器零继续执行)

3. 浮点数的精度: 与尾数的位数和是否规格化有关

4. 浮点数的规格化

① 方法一: 尾数最高位为1 ($r=2$)

② 方法二: 规格化尾数的小数点后第一位总是1, 故规定第一位默认的“1”不明显表示出来。这样可用 n 个数位表示 $n+1$ 位尾数



5. 实际上在计算机中并不使用定点小数表示



定点整数用补码表示; 浮点数用IEEE 754规范标准表示

6. 浮点数IEEE 754标准: 单精度SP (float) 和双精度DP (double)

7. 算术移位(对有符号数的移位)规则: 符号位不变; 补无意义的0或1

4. 算术移位和逻辑移位的区别

6.3

算术移位 有符号数的移位

逻辑移位 无符号数的移位

逻辑左移 低位添 0，高位移丢

逻辑右移 高位添 0，低位移丢

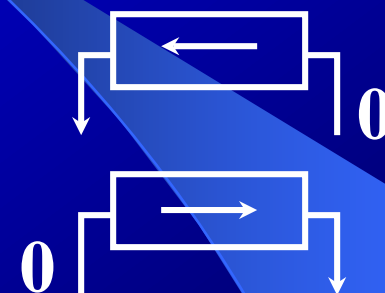
例如 01010011

逻辑左移 10100110

逻辑右移 01011001

算术左移 00100110

算术右移 11011001 (补/反码)



高位 1 移丢

若带进位，算术左移后，符号位 0 移到 C_y ，1 不丢失

进位

C_y

01010011

0

10100110