计算机组成原理

第十六讲

刘松波

哈工大计算学部

模式识别与智能系统研究中心

第 6 章　计算机的运算方法

6.1　无符号数和有符号数

6.2　数的定点表示和浮点表示

6.3　定点运算

6.4　浮点四则运算

6.5　算术逻辑单元

二、浮点表示

6.2

$N = S \times r^j$　　浮点数的一般形式

S 尾数　j 阶码　r 基数（基值）

计算机中 r 取 2、4、8、16 等

当 $r = 2$　　$N = 11.0101$

二进制表示

✓ $= 0.110101 \times 2^{10}$　规格化数

$= 1.10101 \times 2^{1}$

$= 1101.01 \times 2^{-10}$

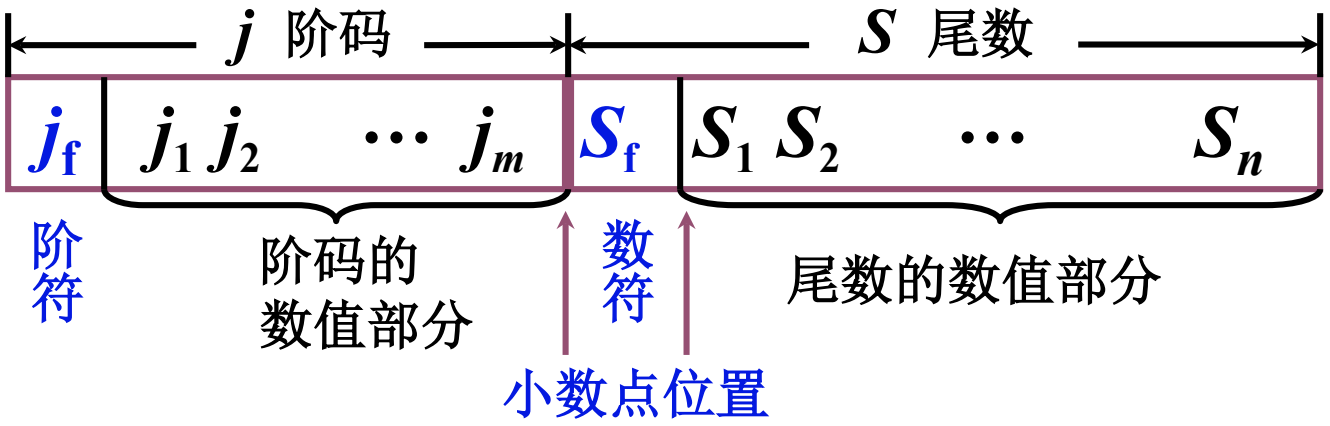
✓ $= 0.00110101 \times 2^{100}$

计算机中　S　小数、可正可负

j　整数、可正可负

1. 浮点数的表示形式

6.2

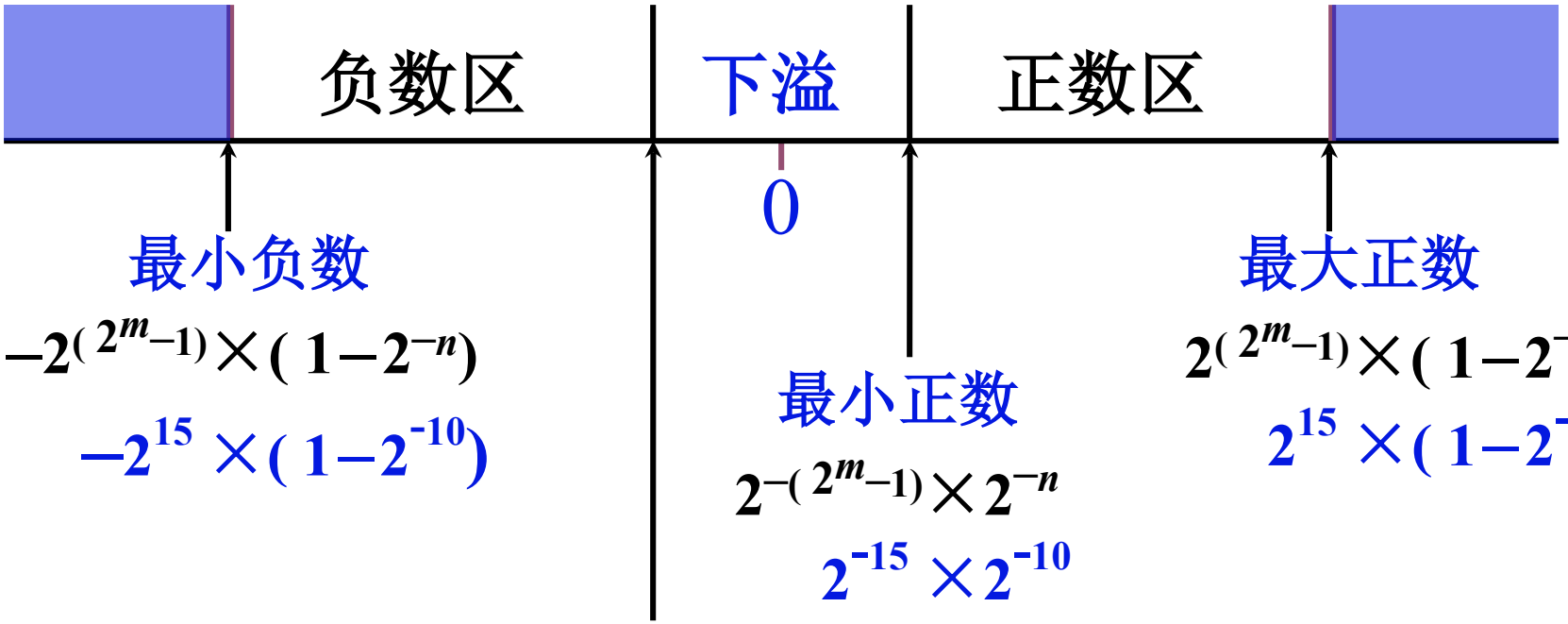


S_f	代表浮点数的符号
n	其位数反映浮点数的精度
m	其位数反映浮点数的表示范围
j_f 和 m	共同表示小数点的实际位置

2. 浮点数的表示范围

上溢　阶码 > 最大阶码

下溢　阶码 < 最小阶码　按 机器零 处理



练习 6.2

设机器数字长为 24 位，欲表示±3万的十进制数，试问在保证数的最大精度的前提下，除阶符、数符各取1 位外，阶码、尾数各取几位？

解：∵ $2^{14}=16384$ $2^{15}=32768$

∴ 如果是定点数15 位二进制数可反映 ±3 万之间的十进制数

$$2^{15}\times 0.\underbrace{\times\times\times\ \cdots\ \times\times\times}_{n\text{位}}$$

$m=4,\ 5,\ 6,\ \cdots$

满足 最大精度 可取 $m=4,\ n=18$

3. 浮点数的规格化形式

$r=2$　　尾数最高位为 1

$r=4$　　尾数最高 2 位不全为 0

$r=8$　　尾数最高 3 位不全为 0

基数不同，浮点数的规格化形式不同

4. 浮点数的规格化

$r=2$	左规	尾数左移 1 位，阶码减 1
	右规	尾数右移 1 位，阶码加 1
$r=4$	左规	尾数左移 2 位，阶码减 1
	右规	尾数右移 2 位，阶码加 1
$r=8$	左规	尾数左移 3 位，阶码减 1
	右规	尾数右移 3 位，阶码加 1

基数 r 越大，可表示的浮点数的范围越大

基数 r 越大，浮点数的精度降低

例如：设 $m=4$，$n=10$，$r=2$

尾数规格化后的浮点数表示范围

最大正数　$2^{+1111} \times 0.\underbrace{1111111111}_{10\text{个}1} = 2^{15} \times (1-2^{-10})$

最小正数　$2^{-1111} \times 0.\underbrace{1000000000}_{9\text{个}0} = 2^{-15} \times 2^{-1} = 2^{-16}$

最大负数　$2^{-1111} \times (-0.\underbrace{1000000000}_{9\text{个}0}) = -2^{-15} \times 2^{-1} = -2^{-16}$

最小负数　$2^{+1111} \times (-0.\underbrace{1111111111}_{10\text{个}1}) = -2^{15} \times (1-2^{-10})$

三、举例

例 6.13　将 $+\frac{19}{128}$ 写成二进制定点数、浮点数及在定点机和浮点机中的机器数形式。其中数值部分均取 10 位，数符取 1 位，浮点数阶码取 5 位（含1位阶符）。

解：　设 $x=+\frac{19}{128}$

二进制形式　　$x=0.0010011$

定点表示　　$x=0.0010011000$

浮点规格化形式　$x=0.1001100000\times2^{-10}$

定点机中　　$[x]_{原}=[x]_{补}=[x]_{反}=0.0010011000$

浮点机中　　$[x]_{原}=1,0010;\ 0.1001100000$

$[x]_{补}=1,1110;\ 0.1001100000$

$[x]_{反}=1,1101;\ 0.1001100000$

例 6.14 将 –58 表示成二进制定点数和浮点数，并写出它在定点机和浮点机中的三种机器数及阶码为移码、尾数为补码的形式（其他要求同上例）。

解： 设 $x = -58$

二进制形式 $x = -111010$

定点表示 $x = -0000111010$

浮点规格化形式 $x = -(0.1110100000) \times 2^{110}$

定点机中

$[x]_{原} = 1, 0000111010$

$[x]_{补} = 1, 1111000110$

$[x]_{反} = 1, 1111000101$

浮点机中

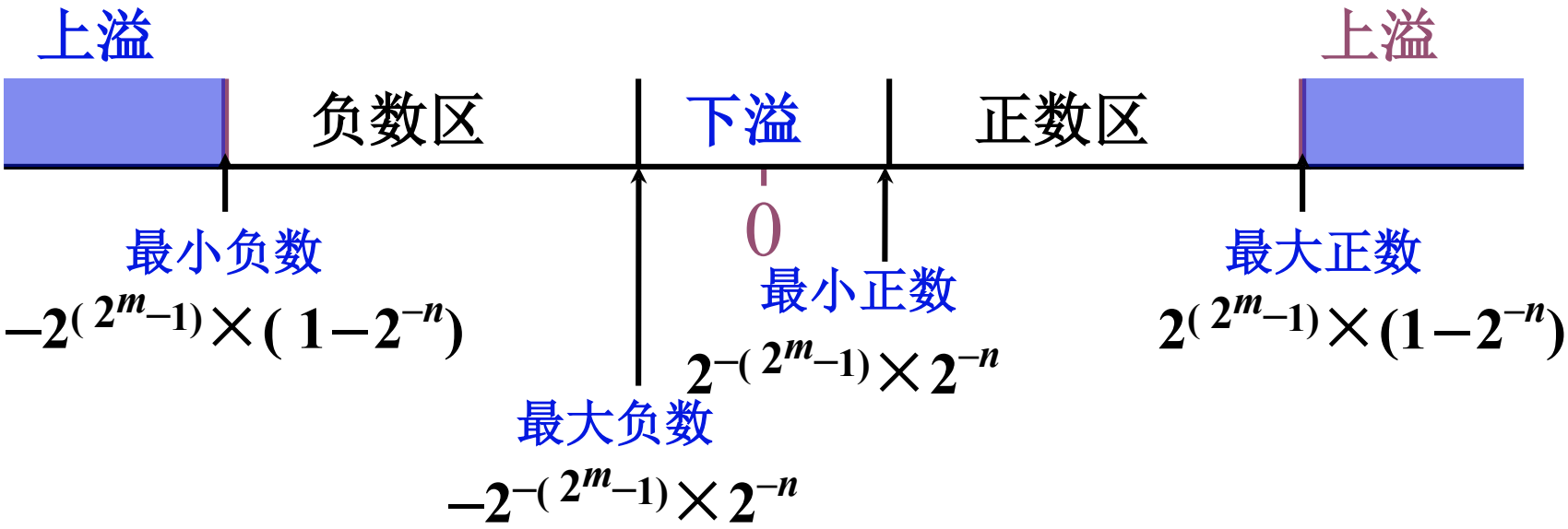
$[x]_{原} = 0, 0110; 1.1110100000$

$[x]_{补} = 0, 0110; 1.0001100000$

$[x]_{反} = 0, 0110; 1.0001011111$

$[x]_{阶移、尾补} = 1, 0110; 1.0001100000$

例6.15　写出对应下图所示的浮点数的补码形式。设 $n=10$，$m=4$，阶符、数符各取 1位。



解：

	真值	补码
最大正数	$2^{15}\times(1-2^{-10})$	0,1111; 0.1111111111
最小正数	$2^{-15}\times 2^{-10}$	1,0001; 0.0000000001
最大负数	$-2^{-15}\times 2^{-10}$	1,0001; 1.1111111111
最小负数	$-2^{15}\times(1-2^{-10})$	0,1111; 1.0000000001

机器零

6.2

- 当浮点数 尾数为 0 时，不论其阶码为何值按机器零处理
- 当浮点数 阶码等于或小于它所表示的最小数 时，不论尾数为何值，按机器零处理

如 $m = 4$　　$n = 10$

当阶码和尾数都用补码表示时，机器零为

×, × × × ×；　0. 0 0　…　0

（阶码 = −16）　1, 0 0 0 0；　×.××　…　×

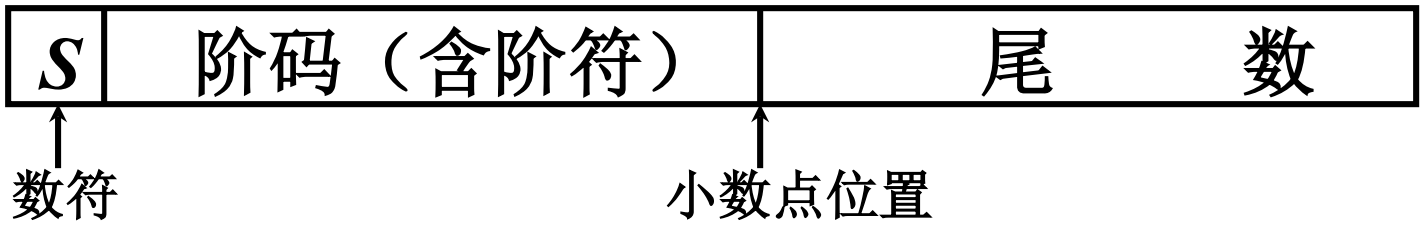
当阶码用移码，尾数用补码表示时，机器零为

0, 0 0 0 0；0. 0 0　…　0

有利于机器中“ 判 0 ” 电路的实现

四、IEEE 754 标准

6.2



尾数为规格化表示

非“0”的有效位最高位为“1”（隐含）

	符号位 *S*	阶码	尾数	总位数
短实数	1	8	23	32
长实数	1	11	52	64
临时实数	1	15	64	80

6.3 定 点 运 算

一、移位运算

1. 移位的意义

15. m = 1500. 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

符号位不变

	码　制	添补代码
正数	原码、补码、反码	0
负数	原　码	0
	补　码	左移 添 0
		右移 添 1
	反　码	1

例6.16

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 $[A]_{原} = [A]_{补} = [A]_{反} = 0,0011010$

移位操作	机　器　数 $[A]_{原}=[A]_{补}=[A]_{反}$	对应的真值
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6.17

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

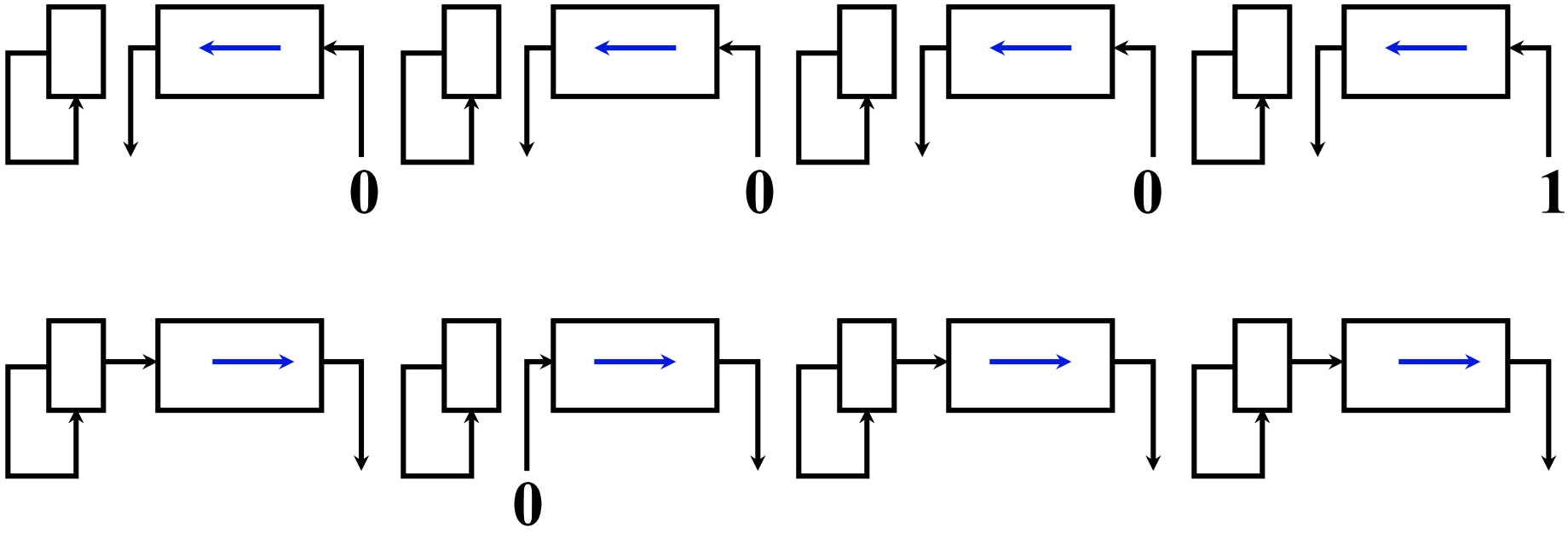
移位操作	机　器　数	对应的真值
移位前	1,1100110	– 26
左移一位	1,1001100	– 52
左移两位	1,0011000	– 104
右移一位	1,1110011	– 13
右移两位	1,1111001	– 7

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	出错	正确	正确
→丢 1	影响精度	影响精度	影响精度	正确

4. 算术移位和逻辑移位的区别

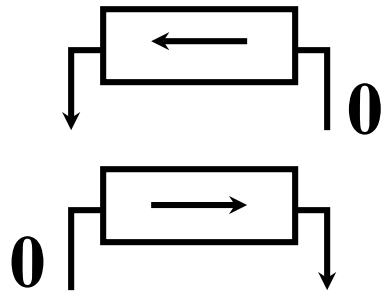
6.3

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　　10110010

逻辑左移　　10100110　　逻辑右移　　01011001

算术左移　　00100110　　算术右移　　11011001（补码）

高位 1 移丢