第6章 计算机的运算方法 习题与解答

> 课件原题 34 道（Slide 259–370 中全部例题、练习、思考题，100% 收录）+ 考点补充题 9 道，共 43 道。
> 所有数值题均经逐步重算验证。竖式中"→1"表示右移/左移一位，逗号/小数点前为符号位。

一、课件原题

题1（课件 p.264，例 6.1）

已知 [x]原 = 1.0011，求 x。

解答：符号位为 1，x 为负小数。由原码定义 [x]原 = 1 − x，得 x = 1 − [x]原 = 1.0000 − 1.0011 = **−0.0011**。

题2（课件 p.264，例 6.2）

已知 [x]原 = 1,1100，求 x。

解答：符号位 1，负整数。x = 2^4 − [x]原 = 10000 − 11100 = **−1100**（即 −12）。

题3（课件 p.265，例 6.3）

已知 [x]原 = 0.1101，求 x。

解答：符号位 0，正数原码即真值：x = **+0.1101**。

题4（课件 p.265，例 6.4）

求 x = 0 的原码。

解答：设 x = +0.0000，则 [+0.0000]原 = 0.0000；设 x = −0.0000，则 [−0.0000]原 = 1 − (−0.0000) = 1.0000。同理整数：[+0]原 = 0,0000，[−0]原 = 1,0000。结论：**[+0]原 ≠ [−0]原，原码中 0 有两种表示**。

题5（课件 p.273，例 6.5）

已知 [x]补 = 0.0001，求 x。

解答：符号位 0，正数补码即真值：x = **+0.0001**。

题6（课件 p.273，例 6.6）

已知 [x]补 = 1.0001，求 x 及 [x]原。

解答：符号位 1，负数。由定义 $x = [x]_{补} - 2 = 1.0001 - 10.0000 = \mathbf{-0.1111}$。求原码：补码数值位 0001 取反得 1110、末位加 1 得 1111，故 $[x]_{原} = \mathbf{1.1111}$。

题7（课件 p.274，例 6.7）

已知 $[x]_{补} = 1,1110$，求 x 及 $[x]_{原}$。

解答：$x = [x]_{补} - 2^5 = 11110 - 100000 = \mathbf{-0010}$（−2）。补码数值位 1110 取反 0001 加 1 得 0010，故 $[x]_{原} = \mathbf{1,0010}$。验证：负数原码→补码同规则互逆，一致。

题8（课件 p.275，练习）

求下列真值的补码（并对照原码）：x = +70、−70、+0.1110、−0.1110、+0.0000、−0.0000、−1.0000。

解答：

真值	[x]补	[x]原
+70 = +1000110	0,1000110	0,1000110
−70 = −1000110	1,0111010	1,1000110
+0.1110	0.1110	0.1110
−0.1110	1.0010	1.1110
+0.0000	0.0000	0.0000
−0.0000	0.0000	1.0000
−1.0000	1.0000	不能表示

步骤：−70：数值 1000110 取反 0111001、加 1 = 0111010 → 1,0111010；−0.1110：1110 取反 0001 加 1 = 0010 → 1.0010；$[-1]_{补} = 2 + (-1.0000) = 10.0000 - 1.0000 = 1.0000$。结论：**[+0]补 = [−0]补**；补码能表示 −1 而原码不能。

题9（课件 p.278，例 6.8）

已知 $[x]_{反} = 0,1110$，求 x。

解答：符号位 0，正数反码即真值：$x = \mathbf{+1110}$。

题10（课件 p.278，例 6.9）

已知 $[x]_{反} = 1,1110$，求 x。

解答：$x = [x]_{反} - (2^5 - 1) = 11110 - 11111 = \mathbf{-0001}$。（也可：数值位 1110 取反 = 0001，加负号。）

题11（课件 p.278，例 6.10）

求 0 的反码。

解答：[+0.0000]反 = 0.0000，[−0.0000]反 = 1.1111；整数 [+0]反 = 0,0000，[−0]反 = 1,1111。**[+0]反 ≠ [−0]反**。

题12（课件 p.280，例 6.11）

设机器数字长 8 位（含 1 位符号位），整数代码分别按无符号数、原码、补码、反码解释时，真值范围各为多少?

解答：

码制	真值范围
无符号数	0 ~ 255
原码	−127 ~ +127
补码	−128 ~ +127
反码	−127 ~ +127

关键代码对照：00000000 → 0 / +0 / +0 / +0；01111111 → 127 / +127 / +127 / +127；10000000 → 128 / −0 / **−128** / −127；10000001 → 129 / −1 / −127 / −126；11111111 → 255 / −127 / −1 / −0。理由：补码 10000000 按定义真值 = 128 − 256 = −128；反码 10000000 数值取反 = 1111111 = 127 故 −127。

题13（课件 p.281，例 6.12）

已知 [y]补，求 [−y]补。

解答：结论：**[y]补 连同符号位在内每位取反、末位加 1，即得 [−y]补**。 推导：情形 I [y]补 = $0.y_1y_2...y_n$（$y \geq 0$）：$-y = -0.y_1...y_n$，$[-y]补 = 2 + (-0.y_1...y_n) = 1.(y_1...y_n$ 取反) + 2^{-n}，恰为 [y]补 全位取反加 1。情形II [y]补 = $1.y_1...y_n$（$y < 0$）：$y = -(0.(y_1...y_n$ 取反) + $2^{-n})$，故 $-y = +0.$(取反) + 2^{-n}，[−y]补 = 0.(取反) + 2^{-n}，同样等于 [y]补 全位取反加 1。例：[y]补 = 0.1011 → 取反 1.0100 加 1 = 1.0101 = [−0.1011]补 ✓。

题14（课件 p.291，练习）

机器数字长 24 位，欲表示 ±3 万的十进制数，在保证最大精度前提下，除阶符、数符各 1 位外，阶码、尾数各取几位?

解答：$2^{14} = 16384 < 30000 < 32768 = 2^{15}$，故规格化浮点数 0.1××...××2^j 需 j 最大取 15（二进制 1111），即阶码数值位 m 满足 2^m − 1 ≥ 15 → m = 4。尾数数值位 n = 24 − 1（阶符）− 1（数符）− 4 = **18**。即 **m = 4，n = 18**（m 再大则 n 减小、精度下降）。

题15（课件 p.294，例 6.13）

将 +19/128 写成二进制定点数、浮点数及定点机/浮点机中的机器数。数值部分 10 位、数符 1 位、阶码 5 位（含 1 位阶符）。

解答：19 = 10011，19/128 = 0.0010011。 - 定点表示：x = 0.0010011000；正数三码相同：[x]原 = [x]补 = [x]反 = **0.0010011000**。 - 浮点规格化：x = 0.1001100000 × 2^{-10}（阶码真值 −2）。 - 阶码 −2 的三种机器数（5 位含阶符）：[−2]原 = 1,0010；[−2]补 = 1,1110；[−2]反 = 1,1101。尾数为正，三码均 0.1001100000。 - 浮点机中：[x]原 = **1,0010; 0.1001100000**；[x]补 = **1,1110; 0.1001100000**；[x]反 = **1,1101; 0.1001100000**。

题16（课件 p.295，例 6.14）

将 −58 表示成二进制定点数和浮点数，写出定点机/浮点机中三种机器数及"阶码移码、尾数补码"形式（位数要求同例 6.13）。

解答：−58 = −111010。 - 定点机：[x]原 = **1,0000111010**；[x]反 = **1,1111000101**；[x]补 = **1,1111000110**（数值 0000111010 取反 1111000101，加 1 得补码数值）。 - 浮点规格化：x = $-0.1110100000 \times 2^{110}$（阶码真值 +6）。阶码 +6 三码均 0,0110。 - 尾数 −0.1110100000：原 1.1110100000；反 1.0001011111；补 1.0001100000。 - 浮点机：[x]原 = **0,0110; 1.1110100000**；[x]补 = **0,0110; 1.0001100000**；[x]反 = **0,0110; 1.0001011111**。 - 阶移尾补：[+6]移 = 2^4 + 6 = 10000 + 0110 = 1,0110 → **1,0110; 1.0001100000**。

题17（课件 p.296，例 6.15）

n = 10、m = 4，阶符数符各 1 位，写出浮点表示范围四个边界值的补码形式。

解答：

真值	补码（阶;尾）
最大正数 $2^{15}\times(1-2^{-10})$	0,1111; 0.1111111111
最小正数 $2^{-15}\times2^{-10}$	1,0001; 0.0000000001
最大负数 $-2^{-15}\times2^{-10}$	1,0001; 1.1111111111
最小负数 $-2^{15}\times(1-2^{-10})$	0,1111; 1.0000000001

步骤：阶码 ±15：[+15]补 = 0,1111，[−15]补 = 1,0001；尾数 −0.0000000001 的补码 = 1.1111111111；−0.1111111111 的补码 = 1.0000000001（取反加 1）。

题18（课件 p.301，例 6.16）

8 位（含 1 符号位），A = +26，写出三种机器数左/右移 1、2 位后的形式与真值，并分析正确性。

解答：正数三码相同：[A]原 = [A]补 = [A]反 = 0,0011010。

操作	机器数	真值
移位前	0,0011010	+26
左移 1 位	0,0110100	+52
左移 2 位	0,1101000	+104
右移 1 位	0,0001101	+13
右移 2 位	0,0000110	+6

分析：左移未丢 1，结果正确（×2、×4）；右移 2 位丢低位 1（26/4 = 6.5），只影响精度，得 +6。

题19（课件 p.302–303，例 6.17）

同上条件，A = −26，写出三种机器数移位结果与真值并分析。

解答：[A]原 = 1,0011010，[A]补 = 1,1100110，[A]反 = 1,1100101。

操作	原码	真值	补码	真值	反码	真值
移位前	1,0011010	−26	1,1100110	−26	1,1100101	−26
左移 1	1,0110100	−52	1,1001100	−52	1,1001011	−52
左移 2	1,1101000	−104	1,0011000	−104	1,0010111	−104
右移 1	1,0001101	−13	1,1110011	−13	1,1110010	−13
右移 2	1,0000110	−6	1,1111001	**−7**	1,1111001	−6

分析：左移正确；右移 2 位时补码添 1 向 −∞ 舍入得 −7，原码/反码得 −6；同一代码 1,1111001 在补码与反码下真值不同。丢 1 影响精度。

题20（课件 p.307，例 6.18）

A = 0.1011，B = −0.0101，求 [A+B]补。

解答：[A]补 = 0.1011，[B]补 = 1.1011。 0.1011 + 1.1011 = 10.0110，符号位进位自然丢掉 → [A+B]补 = **0.0110**。验证：0.1011 − 0.0101 = 0.0110 ✓，A+B = +0.0110。

题21（课件 p.307，例 6.19）

A = −9，B = −5，求 [A+B]补。

解答：A = −1001，B = −0101。[A]补 = 1,0111，[B]补 = 1,1011。 1,0111 + 1,1011 = 11,0010 → 丢进位 → [A+B]补 = **1,0010**，真值 −1110 = −14。验证 −9 + (−5) = −14 ✓（双符号 11 未溢出）。

题22（课件 p.308，例 6.20）

8 位（含 1 符号位），A = 15，B = 24，用补码求 A − B。

解答：[A]补 = 0,0001111，[B]补 = 0,0011000 → [−B]补 = 1,1101000（全位取反加 1）。 0,0001111 + 1,1101000 = **1,1110111** = [A−B]补。真值：数值取反加 1 = 0001001 → A−B = −1001 = **−9** ✓。异号相加不会溢出。

题23（课件 p.308，练习 1）

设 x = 9/16，y = 11/16，用补码求 x + y。

解答：x = 0.1001，y = 0.1011；[x]补 = 0.1001，[y]补 = 0.1011。 0.1001 + 0.1011 = **1.0100**。按补码读作 −0.1100 = −12/16——**课件标"错"**：两正数相加结果为负，发生**溢出**（双符号验证：00.1001 + 00.1011 = 01.0100，01 正溢）。真值 x+y = 20/16 = 1.25，超出小数补码最大 0.1111 = 15/16，故机器结果错误。

题24（课件 p.308，练习 2）

8 位（含 1 符号位），A = −97，B = +41，用补码求 A − B。

解答：[A]补 = 1,0011111，[B]补 = 0,0101001 → [−B]补 = 1,1010111。 1,0011111 + 1,1010111 = 10,1110110 → 丢进位 → 0,1110110 = +118——**课件标"错"**：两负数相加得正，**溢出**（V = 符号位进位 1 ⊕ 最高数值位进位 0 = 1；双符号 11,0011111 + 11,1010111 = 10,1110110 → 10 负溢）。真值 −97 − 41 = −138 < −128，超出表示范围。

题25（课件 p.318–319，例 6.21）

x = −0.1110，y = 0.1101，求 [x·y]原（原码一位乘）。

解答：x* = 0.1110，y* = 0.1101；符号 = 1⊕0 = 1。数值部分竖式（部分积 | 乘数）：

```
  0.0000   1101   初态 z0=0
+ 0.1110          末位1，+x*
  0.1110
→ 0.0111   0110   逻辑右移，得 z1
+ 0.0000          末位0，+0
  0.0111
→ 0.0011   1011   右移，得 z2
+ 0.1110          末位1，+x*
  1.0001
→ 0.1000   1101   右移，得 z3
+ 0.1110          末位1，+x*
  1.0110
→ 0.1011   0110   右移，得 z4
```

$x \cdot y$ = 0.10110110，故 [x·y]原 = **1.10110110**。验证：(14/16)×(13/16) = 182/256 = 0.10110110 ✓。特点：绝对值运算、逻辑右移、移 4 次加 4 次。

题26（课件 p.323–324，例 6.22）

x = 0.111111，y = −0.111001，求 [x·y]原（原码两位乘）。

解答：x* = 0.111111；[x]*补 = 000.111111，[2x*]补 = 001.111110，[−x*]补 = 111.000001（三位符号位）。符号 = 0⊕1 = 1。竖式（部分积 | 乘数 | Cj）：

```
  000.000000   111001   Cj=0  初态
+ 000.111111                  判断01,Cj=0 → +x*
  000.111111
→ 000.001111   111110   Cj=0  补码右移2位
+ 001.111110                  判断10,Cj=0 → +2x*
  010.001101
→ 000.100011   011111   Cj=0  右移2位
+ 111.000001                  判断11,Cj=0 → -x*
  111.100100
→ 111.111001   000111   Cj=1  右移2位
+ 000.111111                  Cj=1 → +x* 校正，不移位
  000.111000   000111
```

$x \cdot y$ = 0.111000000111，故 [x·y]原 = **1.111000000111**。验证：63×57 = 3591，3591/4096 = 0.111000000111 ✓。n=6：右移 3 次（n/2），加 4 次（n/2+1）。

题27（课件 p.325，思考）

n 为奇数时，原码两位乘移几次？最多加几次？

解答：乘数高位补 0 凑成 n+1 位（偶数），共 (n+1)/2 组 → **移 (n+1)/2 次**；每组至多 1 次加法，若结束时 Cj=1 还需 1 次校正加 → **最多加 (n+1)/2 + 1 次**。

题28（课件 p.329，例 6.23）

x = +0.0011，y = −0.1011，求 [x·y]补（Booth 算法）。

解答：[x]补 = 00.0011，[−x]补 = 11.1101，[y]补 = 1.0101，附加位 y_5 = 0。竖式（部分积 | 乘数串 附加位）：

```
  00.0000   1.0101 0   判断10 → +[-x]补
+ 11.1101
  11.1101
→ 11.1110   1 1010 1   补码右移；判断01 → +[x]补
+ 00.0011
  00.0001
→ 00.0000   11 101 0   右移；判断10 → +[-x]补
+ 11.1101
  11.1101
→ 11.1110   111 10 1   右移；判断01 → +[x]补
+ 00.0011
  00.0001
→ 00.0000   1111 1 0   右移；判断10 → +[-x]补
+ 11.1101
  11.1101   1111       最后一步不移位
```

[x·y]补 = **1.11011111**。验证：x·y = (3/16)×(−11/16) = −33/256；1.11011111 真值 = −0.00100001 = −33/256 ✓。共 5 步（n+1），末步不移位。

题29（课件 p.335–336，例 6.24）

x = −0.1011，y = −0.1101，求 [x/y]原（恢复余数法）。

解答：商符 = 1⊕1 = 0；x* = 0.1011，[y]补 = 1.0011。竖式（余数 | 商）：

```
  0.1011   0.0000   +[-y*]补
+ 1.0011
  1.1110   0        余负，上商0（第一次商0，不溢出）
+ 0.1101           恢复余数
  0.1011
→ 1.0110   0        逻辑左移
+ 1.0011           +[-y*]补
  0.1001   01       余正，上商1
→ 1.0010   01       左移
+ 1.0011
  0.0101   011      余正，上商1
→ 0.1010   011      左移
+ 1.0011
  1.1101   0110     余负，上商0
+ 0.1101           恢复余数
  0.1010
→ 1.0100   0110     左移
+ 1.0011
  0.0111   01101    余正，上商1
```

商 = 0.1101，余数 = 0.0111×2⁻⁴；[x/y]原 = **0.1101**。上商 5 次（n+1）、左移 4 次（n）。验证：11/16 ÷ 13/16 = 11/13 ≈ 0.846，0.1101 = 0.8125，余 7/256 = 0.0111×2⁻⁴ ✓。

题30（课件 p.338–339，例 6.25）

同题 29 数据，用加减交替法（不恢复余数）求 [x/y]原。

解答：规则：余正上商 1、下一步 2R−y；*余负上商 0、下一步 2R+y*。

```
  0.1011   0.0000   +[-y*]补
+ 1.0011
  1.1110   0        余负上商0
→ 1.1100   0        左移（不恢复）
+ 0.1101            +[y*]补
  0.1001   01       余正上商1
→ 1.0010   01       左移
+ 1.0011            +[-y*]补
  0.0101   011      余正上商1
→ 0.1010   011      左移
+ 1.0011            +[-y*]补
  1.1101   0110     余负上商0
→ 1.1010   0110     左移
+ 0.1101            +[y*]补
  0.0111   01101    余正上商1
```

商 0.1101、余数 0.0111×2⁻⁴，商符 0 → [x/y]原 = **0.1101**。移 4 次、加 5 次、上商 5 次，与恢复余数法结果一致但省去恢复操作。

题31（课件 p.346，例 6.26）

x = −0.1011，y = +0.1101，求 [x/y]补 并还原真值（补码加减交替除）。

解答：[x]补 = 1.0101，[y]补 = 0.1101，[−y]补 = 1.0011。异号 → 第一步加。上商规则：余与除数同号上 1、异号上 0；末位恒置 1。

```
  1.0101   0.0000   异号做加法 +[y]补
+ 0.1101
  0.0010   1        同号上1（商符）
→ 0.0100   1        左移；商1 → +[-y]补
+ 1.0011
  1.0111   10       异号上0
→ 0.1110   10       左移；商0 → +[y]补
+ 0.1101
  1.1011   100      异号上0
→ 1.0110   100      左移；商0 → +[y]补
+ 0.1101
  0.0011   1001     同号上1
→ 0.0110   1001     左移；商1 → +[-y]补
+ 1.0011
  1.1001   10011    末位恒置1
```

[x/y]补 = **1.0011**，真值 x/y = **−0.1101**。验证：−11/13 ≈ −0.8462，−0.1101 = −0.8125，误差 7/208 ≈ 0.034 < 2^{-4} = 0.0625 ✓（末位恒置 1 的精度误差界）。

题32（课件 p.350、353，浮点加法例）

x = 0.1101×2^{01}，y = (−0.1010)×2^{11}，求 x + y（阶码、尾数均双符号位补码）。

解答：[x]补 = 00,01; 00.1101，[y]补 = 00,11; 11.0110。 ① 对阶：[Δj]补 = [jx]补 + [−jy]补 = 00,01 + 11,01 = 11,10（−2）→ x 阶小，Sx 右移 2 位、jx + 2：[Sx]补' = 00.0011（丢低位 01），[x]补' = 00,11; 00.0011。 ② 尾数求和：00.0011 + 11.0110 = 11.1001 → [x+y]补 = 00,11; 11.1001。 ③ 规格化：11.1001 符号 1、第一数位 1 相同 → 非规格化，**左规** 1 位：11.0010，阶码减 1 → 00,10。 结果 [x+y]补 = 00,10; 11.0010，即 **x+y = (−0.1110)×2^{10}**。 说明：对阶丢弃的低位为 01（0 舍 1 入亦舍去），课件此处未舍入；精确值 −3.375 与机器值 −3.5 之差即对阶截断误差。

题33（课件 p.354–355，例 6.27）

x = 0.1101×2^{10}，y = 0.1011×2^{01}，求 x + y（除阶符数符外阶码 3 位、尾数 6 位，双符号位）。

解答：[x]补 = 00,010; 00.110100，[y]补 = 00,001; 00.101100。 ① 对阶：[Δj]补 = 00,010 + 11,111 = 00,001（+1）→ y 阶小，Sy 右移 1 位、jy + 1：[y]补' = 00,010; 00.010110。 ② 尾数求和：00.110100 + 00.010110 = **01.001010** → 双符号 01，尾数溢出。 ③ **右规**：右移 1 位得 00.100101，阶码加 1：00,010 + 1 = 00,011。结果 [x+y]补 = 00,011; 00.100101，即 **x+y = 0.100101×2^{11}**。验证：3.25 + 1.375 = 4.625 = (37/64)×8 ✓（本次对阶无丢位，结果精确）。

题34（课件 p.356–357，例 6.28）

x = (−5/8)×2^{-5}，y = (7/8)×2^{-4}，求 x − y（阶码 3 位、尾数 6 位，双符号位补码）。

解答：$x = -0.101000\times2^{-101}$ → [x]补 = 11,011; 11.011000；$y = 0.111000\times2^{-100}$ → [y]补 = 11,100; 00.111000。 ① 对阶：[Δj]补 = 11,011 + 00,100 = 11,111（−1）→ x 阶小，Sx 右移 1 位、jx + 1：[x]补' = 11,100; 11.101100。 ② 尾数求差：[Sx]补' + [−Sy]补 = 11.101100 + 11.001000 = 110.110100 → 丢最高进位 = **10.110100**，双符号 10，尾数溢出。 ③ 右规：右移 1 位 = 11.011010，阶码 11,100 + 1 = 11,101 → [x−y]补 = 11,101; 11.011010。 ④ 还原：尾数真值 −0.100110，阶真值 −3 → **$x-y = (-0.100110)\times2^{-11} = (-19/32)\times2^{-3}$**。 验证：−5/256 − 14/256 = −19/256 = −(19/32)×2^{-3} ✓；阶码双符号 11，无溢出。

二、考点补充题

题35【补充•考点：移码与补码、真值互求】

机器数字长 8 位（含 1 位符号位）。(1) 已知 [x]移 = 0,1011010，求 x、[x]补、[x]原；(2) 已知 [x]移 = 1,0110100，求 x、[x]补。

解答：(1) 移码按无符号读 = 01011010_2 = 90；x = 90 − 2^7 = 90 − 128 = **−38**。移码符号位取反得补码：[x]补 = **1,1011010**；原码 = 符号 1 + 38 的二进制 0100110 → [x]原 = **1,0100110**。验证：38 = 0100110，取反 1011001 加 1 = 1011010 ✓。 (2) 无符号读 = 10110100_2 = 180；x = 180 − 128 = **+52**；[x]补 = **0,0110100**（52 = 0110100 ✓）。

题36【补充•考点：双符号位（变形补码）溢出判断】

用双符号位补码计算并判断溢出：(1) x = +0.1011，y = +0.1001，求 x+y；(2) x = −0.1011，y = −0.1001，求 x+y。

解答：(1) [x]补' = 00.1011，[y]补' = 00.1001；和 = **01.0100**。双符号 01 → **正溢（上溢）**。真值 0.1011 + 0.1001 = 1.0100 = 1.25 > 最大 0.1111，确溢出。单符号校验：数值位进位 1、符号位进位 0，V = 1⊕0 = 1 ✓。 (2) [x]补' = 11.0101，[y]补' = 11.0111；和 = 110.1100 → 丢进位 = **10.1100**。双符号 10 → **负溢（下溢）**。真值 −1.25 < −1 ✓。单符号校验：数值位进位 0、符号位进位 1，V = 1 ✓。最高符号位 1 代表真正符号（负）。

题37【补充•考点：IEEE 754 格式与转换】

(1) 写出 IEEE 754 短实数、长实数、临时实数的符号位/阶码/尾数位数与总位数；(2) 说明"隐含位"的含义及其对精度的影响；(3) 将 +13.25 写成短实数形式（补充常识：短实数阶码偏置 127，课件未列偏置值）。

解答：(1) 短实数 1/8/23/32；长实数 1/11/52/64；临时实数 1/15/64/80（课件 p.298）。 (2) 尾数规格化后非 0 有效位最高位恒为 1，该 1 **隐含不存储**；故 23 位尾数字段实际提供 24 位精度（长实数 52→53）。 (3) 13.25 = 1101.01 = 1.10101×2^3。S = 0；阶码字段 = 3 + 127 = 130 = 10000010；尾数字段 = 去掉隐含 1 后的 10101 补 0 至 23 位 = 10101000000000000000000。结果：**0 10000010 10101000000000000000000**（十六进制 41540000H）。回验：阶 130−127=3，1.10101×2^3 = 1101.01 = 13.25 ✓。

题38【补充•考点：串行进位与分组跳跃进位时间】

设与非门级延迟 ty、与或非门延迟 1.5ty，d_i、t_i 已形成。求：(1) 4 位串行进位链产生全部进位的时间；(2) 16 位加法器分别用串行、单重分组（4 位/组）、双重分组跳跃进位链产生全部进位的时间；(3) 32 位加法器三种方案的时间。

解答：(1) 每位进位 2ty，4 位共 **8ty**（课件 p.362）；若用 4 位并行进位链则 2.5ty（p.363）。(2) 16 位：串行 2n·ty = **32ty**；单重分组 4 组依次 2.5/5/7.5/**10ty**（p.364）；双重分组 2.5→5→7.5→**10ty**（p.369）。(3) 32 位：串行 **64ty**；单重分组 8 组 → 2.5×8 = **20ty**；双重分组（2 大组×4 小组）仍四级 2.5/5/7.5/**10ty**（p.370，大组间串行但仅 2 大组）。结论：位数越多，双重分组优势越明显；所有时间均自 d_i、t_i 形成后计。

题39【补充•考点：补码规格化判断（含特例）】

判断下列补码尾数是否规格化（r=2）：(1) 0.01101；(2) 1.10101；(3) 1.01101；(4) 0.10000；(5) 1.00000；(6) 1.10000；(7) 若 (6) 的代码按原码解释，是否规格化?

解答：判据：补码**符号位与第一数位不同**为规格化（课件 p.351）。(1) 0.0… 符号 0 = 首位 0 → **否**（需左规）。(2) 1.1… 相同 → **否**。(3) 1.0… 不同 → **是**。(4) 0.1… → **是**。(5) 1.00000 = −1，特例规定**是**规格化数（p.352）。(6) 1.10000 = −1/2，特例规定**不是**规格化数（p.352）。(7) 原码只看第一数位是否为 1：1.10000 原码 = −0.5，第一数位 1 → **是**（原码不论正负首位 1 即规格化）。

题40【补充•考点：浮点机器数表示（阶补尾补）】

设浮点格式：阶码 4 位（含 1 位阶符，补码），尾数 8 位（含 1 位数符，补码）。将 x = −27/64 写成规格化浮点补码形式，并验证规格化。

解答：27/64 = 0.011011 → x = −0.011011 = **-0.11011×2^{-1}**（左规 1 位后已经规格化形式）。阶码真值 −1：[−1]补（4 位）= **1,111**。尾数 −0.1101100：数值 1101100 取反 0010011 加 1 = 0010100 → [S]补 = **1.0010100**。故 [x]补 = **1,111; 1.0010100**。验证规格化：数符 1、第一数位 0，不同 ✓；回验：尾数真值 −0.1101100 = −27/32，乘 2^{-1} 得 −27/64 ✓。

题41【补充•考点：原码一位乘竖式】

x = −0.1101，y = +0.1011，用原码一位乘求 [x·y]原 及真值。

解答：x* = 0.1101，y* = 0.1011，符号 = 1⊕0 = 1。

```
  0.0000   1011   初态
+ 0.1101          末位1 +x*
  0.1101
→ 0.0110   1101   逻辑右移
+ 0.1101          末位1 +x*
  1.0011
→ 0.1001   1110   右移
+ 0.0000          末位0 +0
  0.1001
→ 0.0100   1111   右移
+ 0.1101          末位1 +x*
  1.0001
→ 0.1000   1111   右移，结束
```

x·y = 0.10001111 → [x·y]原 = **1.10001111**，真值 **−0.10001111 = −143/256**。验证：(13/16)×(11/16) = 143/256 ✓（与课件 p.312 笔算结果一致）。

题42【补充•考点：简答（对阶/判溢/恒置1 原理）】

(1) 浮点对阶为什么必须"小阶向大阶看齐"？(2) 原码除法为什么第一次上商可判溢出？(3) 补码除法为什么末位恒置1？

解答（评分要点）：(1) 对阶需移动尾数：右移只丢低位，误差小且可用舍入补救；左移丢高位，会严重改变数值甚至破坏规格化、引起溢出。故让小阶尾数右移、阶码增大去对齐大阶（课件 p.349）。(2) 小数定点除约定 |x| < |y|。第一步做 |x| − |y|：若余数 ≥ 0（上商 1）说明 |x| ≥ |y|，商 ≥ 1 超出小数表示范围 → 溢出；故**第一次上商 1 即溢出、上商 0 才正常**（课件 p.336、339）。(3) 负商按反码形式上商，末位恒置 1 相当于给反码商加 2^{-n} 化为补码商，省去单独校正步骤，且引入的最大误差为 2^{-n}，在允许精度内（课件 p.343、347）。

题43【补充•考点：选择（范围/规格化/Booth/进位时间）】

(1) 8 位补码（含 1 位符号位）整数表示范围是（ ）。A. −127 ~ +127　B. −128 ~ +127　C. −128 ~ +128　D. −127 ~ +128 (2) 下列补码尾数中规格化的是（ ）。A. 0.01101　B. 1.10101　C. 1.01101　D. 1.10000 (3) Booth 算法中判断位 y_iy_{i+1} = 10 时应（ ）。A. +[x]补后右移　B. +[−x]补后右移　C. +0 后右移　D. +[x]补不移位 (4) 16 位加法器采用单重分组跳跃进位（4 位/组），d_i、t_i 形成后产生全部进位需（ ）。A. 8ty　B. 10ty　C. 16ty　D. 32ty

解答：(1) **B**。补码最小 10000000 = −128、最大 01111111 = +127（p.280）。排除 A/D：那是原码/反码范围；C：+128 需 9 位。(2) **C**。补码规格化要求符号位与第一数位不同：C 为 1.0… 符合。A 0.0…、B 1.1… 相同不规格化；D 1.10000 = −1/2 为特例不规格化（p.351–352）。(3) **B**。$y_{i+1}-y_i$ = 0−1 = −1 → +[−x]补，然后右移 1 位（p.328）。A 对应 01；C 对应 00/11；D 错在"不移位"（仅最后一步不移位）。(4) **B**。4 组依次 2.5ty、5ty、7.5ty、10ty（p.364）。A 是 4 位串行时间；D 是 16 位串行时间；C 无依据。