

13015 计算机系统原理

第二章考点解析（上）

13015 计算机系统原理【第二章考点解析】

考点1 R进制数转换成十进制数

解题思路：按权展开

例 2.1 将二进制数 $(10101.01)_2$ 转换成十进制数。

解题技巧：**个位的下标是：0**

二进制	1	0	1	0	1	.	0	1
下标	4	3	2	1	0		-1	-2

解： $(10101.01)_2 = (1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2})_{10} = (21.25)_{10}$

13015 计算机系统原理【第二章考点解析】

考点1 R进制数转换成十进制数

解题思路：按权展开

例 2.2 将八进制数 $(307.6)_8$ 转换成十进制数。

解题技巧：**个位**的下标是：**0**

八进制	3	0	7	.	6
下标	2	1	0		-1

解： $(307.6)_8 = (3 \times 8^2 + 0 \times 8^1 + 7 \times 8^0 + 6 \times 8^{-1})_{10} = (199.75)_{10}$

13015 计算机系统原理【第二章考点解析】

考点1 R进制数转换成十进制数

解题思路：按权展开

例 2.3 将十六进制数 $(3A.C)_{16}$ 转换成十进制数。

解题技巧：个位的下标是：0

十六进制	3	A	.	C
下标	1	0		-1

$$\text{解：} (3A.C)_{16} = (3 \times 16^1 + 10 \times 16^0 + 12 \times 16^{-1})_{10} = (58.75)_{10}$$

13015 计算机系统原理 【第二章考点解析】

考点2 十进制数转换成R进制数

解题思路： **整数**和**小数**分别进行转换

- (1) 整数部分：除基数商为0，逆向取余
- (2) 小数部分：乘基数积为0，正向取整

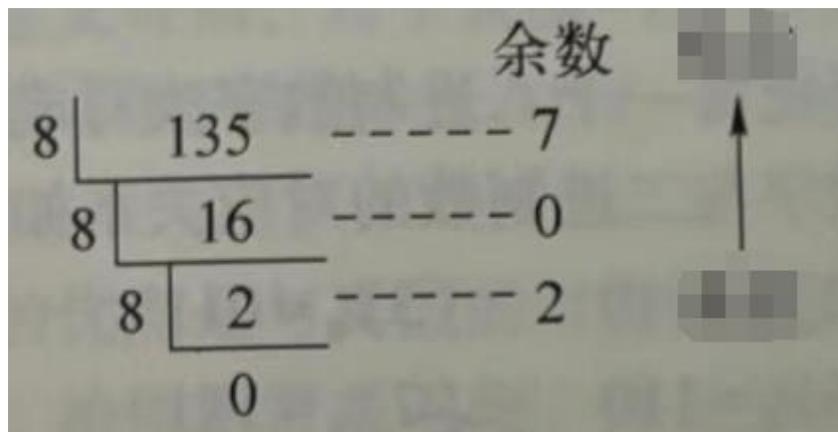
13015 计算机系统原理 【第二章考点解析】

考点2 十进制数转换成R进制数

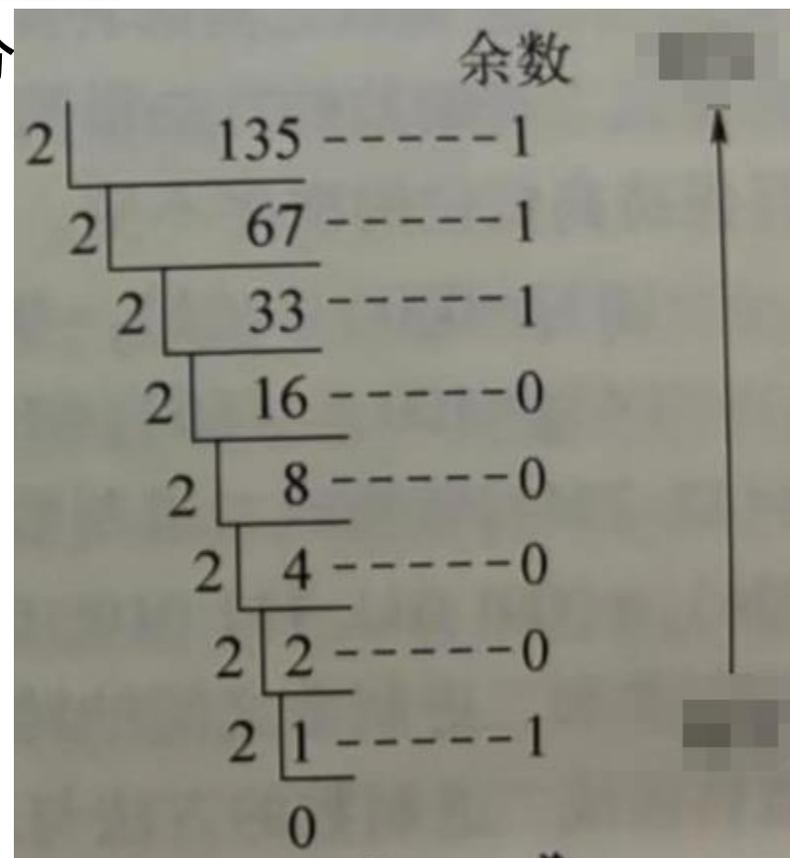
例 2.4 将十进制整数 135 分别转换成八进制数和二进制数。

解题技巧：先看数字组成部分，是否有整数和小数部分

(1) 135是整数，除基数商为0，逆向取余



所以， $(135)_{10} = (207)_8 = (10000111)_2$



13015 计算机系统原理 【第一章考点解析】

考点2 十进制数转换成R进制数

例 2.5 将十进制小数 0.6875 分别转换成二进制数和八进制数。

解题技巧：先看数字组成部分，是否有整数和小数部分

(1) 0.6875是小数，乘基数积为0，正向取整

$$\text{解： } 0.6875 \times 2 = 1.375 \quad \text{整数部分} = 1$$

$$0.375 \times 2 = 0.75 \quad \text{整数部分} = 0$$

$$0.75 \times 2 = 1.5 \quad \text{整数部分} = 1$$

$$0.5 \times 2 = 1.0 \quad \text{整数部分} = 1$$

所以， $(0.6875)_{10} = (0.1011)_2$

$$0.6875 \times 8 = 5.5 \quad \text{整数部分} = 5$$

$$0.5 \times 8 = 4.0 \quad \text{整数部分} = 4$$

所以， $(0.6875)_{10} = (0.54)_8$

13015 计算机系统原理 【第二章考点解析】

考点3 二、八、十六进制数的相互转换

解题思路：

(1) R进制转八、十六进制：

用**二进制**做桥梁，转**八进制**是**三合一**，转**十六进制**是**四合一**，从**←**数，如果位数不够最左边补0

例：二进制：01101010，转成八进制和十六进制数

001|101|010 -> 八进制： 152

0110|1010 -> 十六进制： 6A

(2) 八、十六进制转二进制：

八进制是**一拆三**，**十六进制**是**一拆四**，从**→**数

例：八进制：15，十六进制：26，转成二进制数

八进制：15 -> 二进制： 001 101 → 补位 00001101

十六进制：26 -> 二进制： 0010 0110

13015 计算机系统原理 【第二章考点解析】

考点3 二、八、十六进制数的相互转换

$$\begin{array}{llll} (0)_8 = 000 & (1)_8 = 001 & (2)_8 = 010 & (3)_8 = 011 \\ (4)_8 = 100 & (5)_8 = 101 & (6)_8 = 110 & (7)_8 = 111 \end{array}$$

例 2.8 将 $(13.724)_8$ 转换成二进制数。

解: $(13.724)_8 = (001\ 011.111\ 010\ 100)_2 = (1011.1110\ 101)_2$

例 2.9 将十六进制数 $(2B.5E)_{16}$ 转换成二进制数。

解: $(2B.5E)_{16} = (0010\ 1011.0101\ 1110)_2 = (10\ 1011.0101\ 111)_2$

13015 计算机系统原理 【第二章考点解析】

考点4 定点数的编码表示

机器数：通常将**数值数据**在计算机**内部编码**表示的数

真值：机器数真正的值（现实世界中带有**正负号的数**）

原码：由**符号-数值**表示（**0：正；1：负**）

例：**0110 1100** 或 **1001 0011**

补码：用加法来实现减法运算，也称[**2-补码**表示法]

反码：1- \rightarrow 0; 0- \rightarrow 1

移码：浮点数的阶用[移码]表示

13015 计算机系统原理 【第二章考点解析】

考点4 定点数的编码表示

一、真值(原码)求补码

解题思路：

- 1、判断该真值为正数还是负数；
- 2、若：**正数**，则**符号位取0**，**数值部分**：**不变**；
- 3、若：**负数**，则**符号位取1**，**数值部分**：**各位取反，末位加1**

13015 计算机系统原理 【第二章考点解析】

考点4 定点数的编码表示

一、真值(原码)求补码

例 2.16 假设补码位数为 8，用简便方法求数 -110 0011 的补码表示。

解题思路：

该真值是**负数**，**符号位为1**，**数值部分**：**各位取反，末位加1**

	符号	数值							
	-	1	1	0	0	0	1	1	真值
		0	0	1	1	1	0	0	各位反码
+								1	末位加1
	1	0	0	1	1	1	0	1	补码

解： $[-110\ 0011]_{补} = 1\ 001\ 1100 + 0\ 0000001 = 1\ 001\ 1101$

13015 计算机系统原理 【第二章考点解析】

考点4 定点数的编码表示

二、补码求真值(原码)

解题思路：

- 1、判断符号位；
- 2、若：符号位为0，则真值的符号是正数，数值部分：不变；
- 3、若：符号位为1，则真值的符号是负数，数值部分：各位取反，末位加1

13015 计算机系统原理 【第二章考点解析】

考点4 定点数的编码表示

二、补码求真值(原码)

例 2.17 已知 $[X_T]_{\text{补}} = 1\ 011\ 0100$ ，求真值 X_T 。

解题思路：

该符号位为1，真值是负数，数值部分：各位取反，末位加1

	符号	数值							
	1	0	1	1	0	1	0	0	补码
		1	0	0	1	0	1	1	各位反码
+								1	末位加1
	-	1	0	0	1	1	0	0	真值

解： $X_T = -(100\ 1011 + 1) = -100\ 1100$

13015 计算机系统原理【第二章考点解析】

考点4 定点数的编码表示

三、补码求原码相反数的补码

解题思路：

1、符号位 + 数值位：各位取反，末位加1

区别：

与【真值(原码)求补码】和【补码求真值(原码)】，只有【数值位】各位取反，末位加1

13015 计算机系统原理 【第二章考点解析】

考点4 定点数的编码表示

三、补码求原码相反数的补码

例 2.18 已知 $[X_T]_{补} = 1\ 011\ 0100$ ，求 $[-X_T]_{补}$

解题思路： 符号位 + 数值位：各位取反，末位加1

	符号	数值							
	1	0	1	1	0	1	0	0	补码
	0	1	0	0	1	0	1	1	各位反码
+								1	末位加1
	0	1	0	0	1	1	0	0	相反数补码

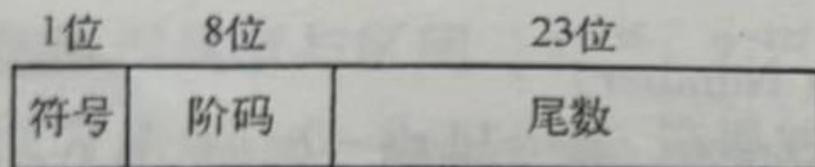
解： $[-X_T]_{补} = 0\ 100\ 1011 + 0\ 000\ 0001 = 0\ 100\ 1100$

13015 计算机系统原理【第二章考点解析】

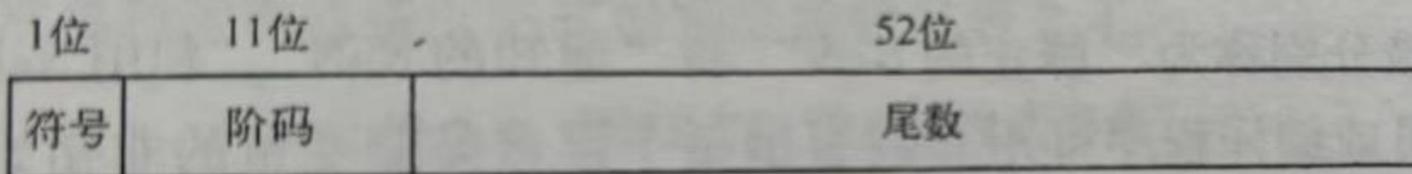
考点5 IEEE754浮点数标准

在这个标准中，提供了两种基本浮点格式：

32位单精度和**64位双精度**格式，其中：主要考察**32位单精度**由三部分组成，符号 + 阶码 + 尾数



a) 32位单精度格式



b) 64位双精度格式

图 2.3 IEEE 754 浮点数格式

13015 计算机系统原理【第二章考点解析】

考点6 进制数转换IEEE754浮点数格式表示

解题思路：

- 1、所求的数如果不是二进制先转为二进制；
- 2、转为二进制的科学计数法，格式为： $1.F \times 2^n$ ；
- 3、确定**符号**，**阶码**和**尾数**；
- 4、符号：**0正，1负**
- 5、阶码： **$n+127$ [偏置常数]**(32位)， **$n+1023$**
- 6、尾数：**F**，后面补0，补够**23**位
- 7、四合一，转为十六进制（如果题目有要求）

13015 计算机系统原理 【第二章考点解析】

考点7 IEEE754浮点数格式表示转换为真值

解题思路：

- 1、转换为IEEE754浮点数格式表示；
- 2、确定**符号**，**阶码**和**尾数**；
- 3、符号：**0：正；1：负**
- 4、阶码：8位阶码转换为十进制，再减127
- 5、尾数：23位尾数转换为0.F格式，二进制转十进制
- 6、由3、4、5可得最终的真值

13015 计算机系统原理【第二章考点解析】

考点7 IEEE754浮点数格式表示转换为

例 2.22 求机器数为 C0A0 0000H 的 IEEE 754 单精度浮点数的值。

解题思路：

- 1、转二进制：1100000010100000...000，这已经是32位单精度浮点数格式；
- 3、确定符号，阶码和尾数 (1+8+23)；
- 4、符号：1：负 (1位)
- 5、阶码：10000001，转为真值：129， $129-127=2$ ，所以是 2^2 ； (8位)
- 6、尾数：0100000...000，小数部分：0.01，二进制转为十进制：0.25；
- 7、最终结果： $-1 \times 1.25 \times 2^2 = -5$

13015 计算机系统原理【第二章考点解析】

考点8 C语言中的浮点数类型

解题思路： int -> float -> double; 表示的**范围越来越大**

小->大，不会溢出，可能舍入

大->小，可能溢出，可能舍入，也可能截断

int表示的范围：-2147483648 ~ 2147483647

float表示的范围：-3.4 × 10³⁸ ~ 3.4 × 10³⁸，**精度**约为**6-7**位有效数字

double表示的范围：-1.7 × 10³⁰⁸ ~ 1.7 × 10³⁰⁸，**精度**约为**15-16**位有效数字

1、int -> float;

小到大，不会溢出，但是因为float精度有效位数少，**有可能舍入**。

例：int 值 123456789 转换为 float 类型，由于float精度限制，可能会舍入为 1.234568 × 10⁸

13015 计算机系统原理【第二章考点解析】

考点8 C语言中的浮点数类型

2、int(或float) -> double;

因为double类型的有效位数更多，故能保留精确值。

3、double -> float;

因为float类型表示范围更小，**可能溢出**，由于有效位数变少，**可能舍入**。

4、float(或double) -> int ;

因为int没有小数部分，所以**可能会截断**，例：1.999被转换为1，因为int表示的范围更小，**可能溢出**

13015 计算机系统原理【第二章考点解析】

考点8 C语言中的浮点数类型

例 2.23 假定变量 i 、 f 、 d 的类型分别是 `int`、`float` 和 `double`，它们可以取除 $+\infty$ 、 $-\infty$ 和 NaN 以外的任意值。请判断下列每个 C 语言关系表达式在 32 位机器上运行时是否永真。

- A. `i == (int) (float) i`
- B. `f == (float) (int) f`
- C. `i == (int) (double) i`
- D. `f == (float) (double) f`
- E. `d == (float) d`
- F. `f == -(-f)`
- G. `(d+f)-d == f`

13015 计算机系统原理 【第二章考点解析】

考点9 数据的宽度和单位

- 1、**比特(bit)**，也叫：位，用**b**表示**比特**
- 2、**字节(Byte)**，一个字节等于8位，用**B**表示**字节**
- 3、**字(Word)**，一般情况下，可以理解为双字节，16位

【引申】第三章中会应用

- 4、**字长**，CPU内部用于整数运算的数据通路的宽度 **【名词解释题】**
平常所说的机器是32位还是64位，指的就是字长

$$1B = 8b$$

$$1KB = 1024B = 2^{10}B$$

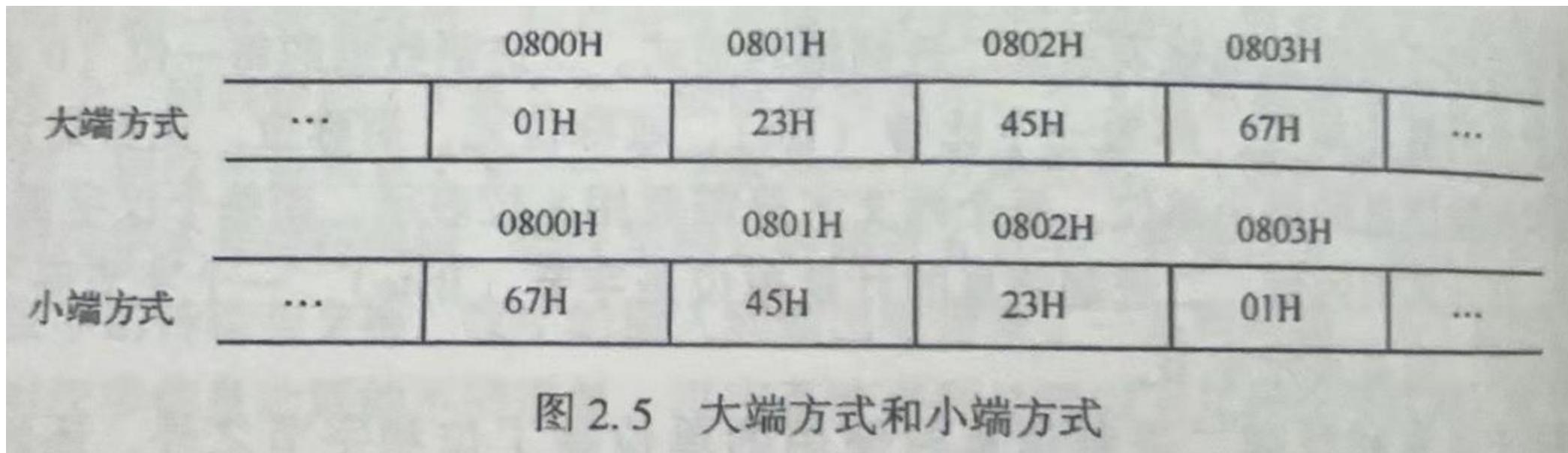
$$1MB = 1024KB = 2^{10}KB$$

$$1GB = 1024MB = 2^{10}MB$$

$$1TB = 1024GB = 2^{10}GB$$

13015 计算机系统原理【第二章考点解析】

考点10 数据的存储和排列顺序



- 1、在按字节编制的计算机中，假定int类型的变量i的**机器数**为**01234567H** **(左高右低)**
- 2、**01H**为MSB表示**最高**有效字节，**67H**为LSB表示**最低**有效字节
- 3、从上图可知**0800H**是**低地址**，**0803H**是**高地址** **(地址：左低右高)**
- 4、助记符：**最高有效字节在最高位为小端方式，高高小(或低低小)**
最高有效字节在最低位为大端方式，高低大(或低高大)

13015 计算机系统原理【第二章考点解析】

考点11 加法和定、浮点数乘除运算

计算机中浮点数加减运算的对接操作是：阶码较小的数，阶码增大，尾数右移

ALU的核心部件是：加法器

乘法是：Booth(布斯)

战略性放弃

因为：不论是计算机组成原理 还是 计算机系统原理，都没有考过

13015 计算机系统原理

第二章考点解析（历年真题讲解）

13015 计算机系统原理【2024-04】

22. 字长

解析：

是指CPU内部用于整数运算的数据通路的宽度。

13015 计算机系统原理【2024-04】

31. 将十进制数 100 转换为 IEEE754 的单精度（32 位）浮点数格式，要求最后格式用十六进制数表示。注：IEEE754 单精度浮点数的计数公式为 $(-1)^s \times 1.f \times 2^{E-127}$ ，其中符号位 1 位，阶码 8 位，尾数 23 位。

解题思路：

- 1、转二进制：100 -> 1100100；
- 2、科学计数法：1.100100 × 2的6次方；
- 3、确定**符号**，**阶码**和**尾数**；
- 4、符号：**正：0**（1位）
- 5、阶码：**6 + 127 = 133**，转为二进制为：**1000 0101**（8位）
- 6、尾数：**100100**，后面补17个0，为：**100100...00**（23位）
- 7、最终结果：**01000010110010000000000000000000**
- 8、转为十六进制：**42C8 0000H**

13015 计算机系统原理【2024-04】

总结:

(1) 名词解释**3分**

(2) 计算题**1×10=10分**

第二章共计13分, 占比13%

13015 计算机系统原理【2023-10】

4. 对于 IEEE754 浮点数表示格式，下列说法正确的是
- A. 阶码用补码表示
 - B. 表示尾数部分的最高位为尾数的符号位
 - C. 阶码用原码表示
 - D. 最高位为浮点数的符号位
5. 已知数 $[X]_{\text{补}}$ 和 $[Y]_{\text{补}}$ ，在运算器中计算 $[X-Y]_{\text{补}}$ 的方法是
- A. $[X]_{\text{补}} + [\overline{Y}]_{\text{补}-1}$
 - B. $[X]_{\text{补}} + [\overline{Y}]_{\text{补}+1}$
 - C. $[X]_{\text{补}} - [\overline{Y}]_{\text{补}-1}$
 - D. $[X]_{\text{补}} - [\overline{Y}]_{\text{补}+1}$
6. 在计算机中浮点数加减运算的对阶操作是
- A. 阶码较小的数，阶码增大，尾数右移
 - B. 阶码较小的数，阶码增大，尾数左移
 - C. 阶码较大的数，阶码减小，尾数左移
 - D. 阶码较大的数，阶码减小，尾数右移

13015 计算机系统原理【2023-10】

解析：

1、正确答案：D

阶码：【移码】，最高位：符号位

2、正确答案：B

战略性放弃

3、正确答案：A

计算机中浮点数加减运算的对接操作是：阶码较小的数，阶码增大，尾数右移

13015 计算机系统原理 【2023-10】

30. 将十进制数 67.375 转换成 IEEE754 的 32 位标准浮点数的二进制格式, 并写出其 16 进制数格式。

解题思路:

- 1、转二进制: $67.375 \rightarrow 1000011.011$ (注意: **整数**和**小数**部分**分别转换**) ;
- 2、科学计数法: 1.000011011×2 的**6**次方;
- 3、确定**符号**, **阶码**和**尾数**;
- 4、符号: **正: 0 (1位)**
- 5、阶码: $6 + 127 = 133$, 转为二进制为: **1000 0101 (8位)**
- 6、尾数: **000011011**, 后面补14个0, 为: **000011011...00 (23位)**
- 7、最终结果: **01000010100001101100000000000000**
- 8、转为十六进制: **4286 C000H**

13015 计算机系统原理【2023-10】

总结:

(1) 选择题 $3 \times 1 = 3$ 分

(2) 计算题 $1 \times 10 = 10$ 分

第二章共计13分, 占比13%

13015 计算机系统原理【2023-04】

1. 十进制数-72 的 8 位补码表示是

A. 01001000

B. 11001000

C. 10110111

D. 10111000

解析：

1、正确答案：D

-72 转二进制 -1001000，负数，数值位各位取反，末位加1，可得：D

13015 计算机系统原理【2023-04】

22. 机器数

解析：

22、将数值数据在计算机内部编码表示的数

13015 计算机系统原理【2023-04】

27. 既然计算机内部所有信息都用二进制表示，为什么还要用到八进制或十六进制数？

解析：

27、答：（1）二进制数太长，**书写、阅读均不方便**，而八进制或十六进制数却像十进制数一样简练、易写易记；（2）虽然计算机中知识用二进制一种计数制，但**为了开发和调式程序、阅读机器代码时的方便**，人们经常用**八进制或十六进制数来等价的表示**二进制数。

13015 计算机系统原理【2023-04】

31. 将十进制数 160 转换为 IEEE754 的单精度 (32 位) 浮点数格式, 要求最后格式用十六进制数表示。注: IEEE754 单精度浮点数的计数公式为 $(-1)^s \times 1.f \times 2^{E-127}$, 其中符号位 1 位, 阶码 8 位, 尾数 23 位。

解题思路:

- 1、转二进制: $160 \rightarrow 10100000$;
- 2、科学计数法: 1.0100000×2^7 ;
- 3、确定**符号**, **阶码**和**尾数**;
- 4、符号: **正: 0 (1位)**
- 5、阶码: $7 + 127 = 134$, 转为二进制为: **1000 0110 (8位)**
- 6、尾数: **0100000**, 后面补16个0, 为: **0100000...00 (23位)**
- 7、最终结果: **01000011001000000000000000000000**
- 8、转为十六进制: **4320 0000H**

13015 计算机系统原理【2023-04】

总结:

(1) 填空题**1分**

(2) 名词解释题**3分**

(3) 简答题**6分**

(4) 计算题 **$1 \times 10 = 10$ 分**

第二章共计20分，占比20%

The background features a blue-toned digital landscape. In the foreground, there are rolling hills or dunes covered in a network of glowing white lines and small dots, suggesting a data or network structure. The sky is a gradient of blue, with several bright, out-of-focus stars or light points scattered across it. The overall aesthetic is clean, modern, and technological.

谢谢大家