13015 计算机系统原理

第2章 数据的表示和运算

13015 计算机系统原理【第二章考点解析】

考点1 信息的二进制编码

在计算机内部，所有信息都用二进制数字表示。

指令所处理的**基本数据**类型分为和两种：

1）数值数据

2）非数值数据

二进制编码的十进制数(Binary Coded Decimal Number，BCD)：**8421BCD**

比如：**$(1010001)_{BCD}$**，每4位代表一个十进制数字，不足位时在高位(左侧)补零：1010001补零为01010001，分组为0101(5)和0001(1)，**$(1010001)_{BCD}=(51)_{10}$**。

13015 计算机系统原理【第二章考点解析】

考点1 信息的二进制编码

2. 某数在计算机中用 8421BCD 码表示为 011110001001，其真值为

A. 789　　B. 789H　　C. 1929　　D. 11110001001B

【答案】：A【2024年04月】

【解析】：011110001001 的每四位对应一个十进制数字

0111 对应 7，1000 对应 8，1001 对应 9，所以其真值为 789

24. 简述指令所处理的基本数据类型的数值数据和非数值数据。

【答案】：【2025年10月】

数值数据可用来表示数量的多少，可比较其大小，分为整数和实数，整数又分为无符号整数和带符号整数。

非数值数据没有大小之分，不表示数量的多少，主要包括字符数据和逻辑数据。

13015 计算机系统原理【第二章考点解析】

考点2 进制记数制

十进制：

2585.62 代表的值是：$(2585.62)_{10}=2\times10^3+5\times10^2+8\times10^1+5\times10^0+6\times10^{-1}+2\times10^{-2}$

二进制：

$(100101.01)_2=1\times2^5+0\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2}$

在计算机系统中，常用的几种进位记数制有下列几种。

二进制　$R=2$，　基本符号为 0 和 1。

八进制　$R=8$，　基本符号为 0,1,2,3,4,5,6,7。

十六进制 $R=16$，基本符号为 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F。

十进制　$R=10$，基本符号为 0,1,2,3,4,5,6,7,8,9。

二进制：**B**　　八进制：**O**　　十进制：**D**　　【十进制数的后缀可省略】

十六进制：**H**　　比如：十六进制数**308FH**或 **0x308F**等

13015 计算机系统原理【第二章考点解析】

考点2 进制记数制

13. 使用后缀字母标识该数的进位数制，一般用_____表示二进制，用_____表示十六进制。

【答案】：B，H【2025年10月】

【解析】：

二进制：B

八进制：O

十进制：D　　【十进制数的后缀可省略】

十六进制：H　　比如：十六进制数308FH或 0x308F等

13015 计算机系统原理【第二章考点解析】

考点3 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：$(10101.01)_2=(1\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2})_{10}=(21.25)_{10}$

二进制	1	0	1	0	1	.	0	1
指数	4	3	2	1	0		−1	−2

13015 计算机系统原理【第二章考点解析】

考点3 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：$(307.6)_8=(3\times8^2+7\times8^0+6\times8^{-1})_{10}=(199.75)_{10}$

八进制	3	0	7	.	6
指数	2	1	0		−1

13015 计算机系统原理【第二章考点解析】

考点3 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：$(3A.C)_{16}=(3\times16^{1}+10\times16^{0}+12\times16^{-1})_{10}=(58.75)_{10}$

十六进制	3	A	.	C
指数	1	0		−1

13015 计算机系统原理【第二章考点解析】

考点3 R进制数转换成十进制数

1. 下列数中最大的数是

A. $(98)_{16}$　　B. $(227)_{8}$

C. $(10011001)_{2}$　　D. $(152)_{10}$

【答案】：C【2024年10月】

【解析】：“按权展开”

A. $9\times16^{1}+8\times16^{0}=152$

B. $2\times8^{2}+2\times8^{1}+7\times8^{0}=151$

C. $1\times2^{7}+1\times2^{4}+1\times2^{3}+1\times2^{0}=153$

D. 152

13015 计算机系统原理【第二章考点解析】

考点3 R进制数转换成十进制数

1. 下列数中最小的数是

A. $(101001)_2$　　B. $(52)_8$

C. $(1010001)_{BCD}$　　D. $(233)_{16}$

【答案】：A【2025年04月】

【解析】：“按权展开”

A: $(101001)_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 32 + 8 + 1 = 41$

B: $(52)_8 = 5 \times 8^1 + 2 \times 8^0 = 40 + 2 = 42$

C: $(1010001)_{BCD}$ 是 BCD（二进制编码的十进制）表示。BCD 每4位代表一个十进制数字，不足位时在左侧补零：1010001 补零为 01010001，分组为 0101（5）和 0001（1），因此值为 51。

D: $(233)_{16} = 2 \times 16^2 + 3 \times 16^1 + 3 \times 16^0 = 512 + 48 + 3 = 563$

13015 计算机系统原理【第二章考点解析】

考点3 R进制数转换成十进制数

3. 下列数中最大的是

A. $(11001)_2$　　B. $(150)_8$　　C. $(110)_{10}$　　D. $(70)_{16}$

【答案】：D【2025年10月】

【解析】：“按权展开”

A. $(11001)_2 = 1\times2^4+1\times2^3+0\times2^2+0\times2^1+1\times2^0 = 16+8+0+0+1 = \mathbf{25}$

B. $(150)_8 = 1\times8^2+5\times8^1+0\times8^0 = 64+40+0 = \mathbf{104}$

C. $(110)_{10} = \mathbf{110}$

D. $(70)_{16} = 7\times16^1+0\times16^0 = \mathbf{112}$

13015 计算机系统原理【第二章考点解析】

考点4 十进制数转换成R进制数

任何一个**十进制**数转换成 R 进制数时，要将**整数**和**小数**部分分别进行转换。

1）**整数**部分的转换

转换方法：**除基商0，逆向取余**

2）**小数**部分的转换

转换方法：**乘基积0，正向取整**

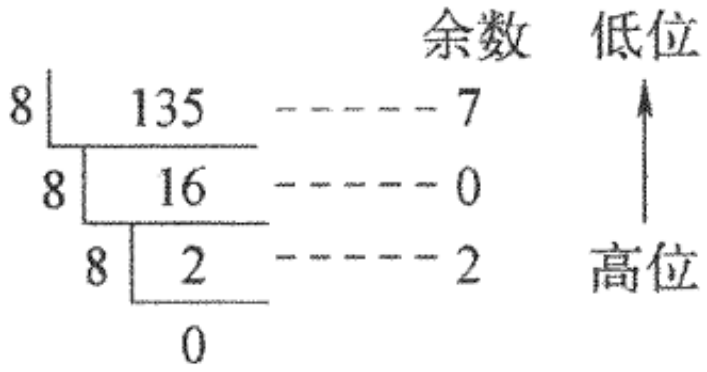
13015 计算机系统原理【第二章考点解析】

考点4 十进制数转换成R进制数

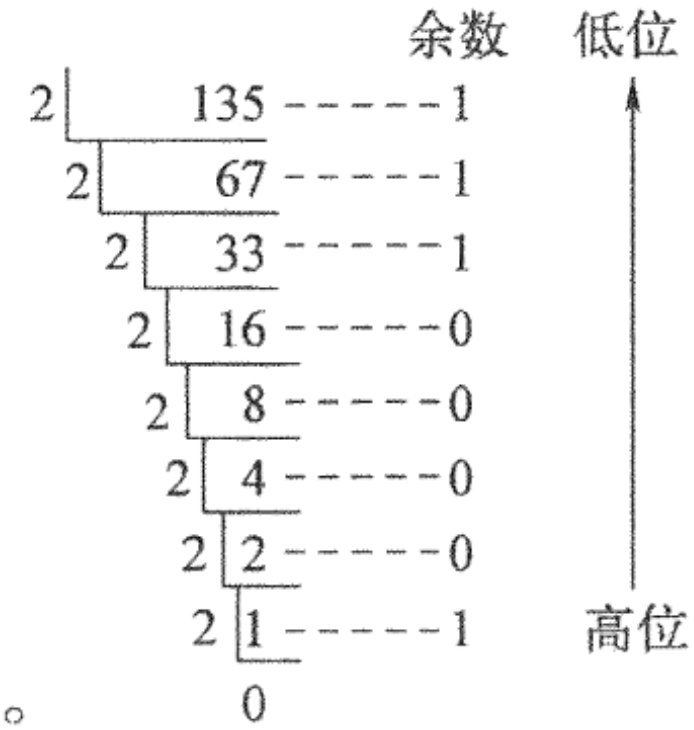
例 2.4　将十进制整数 135 分别转换成八进制数和二进制数。

解题思路：**整数**部分的转换方法：**除基商0，逆向取余**

1）八进制：**基数**是：8



2）二进制：**基数**是：2



所以，$(135)_{10}=(207)_{8}=(10000111)_{2}$。

13015 计算机系统原理【第二章考点解析】

考点4 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。

解题思路：**小数**部分的转换方法：**乘基积0，正向取整**

1）二进制：**基数**是：2

解：
$0.6875\times2=1.375$　整数部分=1　（高位）
$0.375\times2=0.75$　整数部分=0　↓
$0.75\times2=1.5$　整数部分=1　↓
$0.5\times2=1.0$　整数部分=1　（低位）

所以，$(0.6875)_{10}=(0.1011)_2$

2）八进制：**基数**是：8

$0.6875\times8=5.5$　整数部分=5　（高位）
$0.5\times8=4.0$　整数部分=4　（低位）

所以，$(0.6875)_{10}=(0.54)_8$

例 2.7 将十进制数 135.6875 分别转换成二进制数和八进制数。

解：$(135.6875)_{10}=(1000\ 0111.1011)_2=(207.54)_8$

13015 计算机系统原理【第二章考点解析】

考点4 十进制数转换成R进制数

例 2.6　将十进制小数 0.63 转换成二进制数。

解：0.63×2=1.26　　整数部分=1　　（高位）

0.26×2=0.52　　整数部分=0　　↓

0.52×2=1.04　　整数部分=1　　↓

0.04×2=0.08　　整数部分=0　　（低位）

所以，$(0.63)_{10}=(0.1010\cdots)_2$

备注：在转换过程中，可能乘积的**小数部分总得不到 0**，即：转换得到希望的位数后还有余数，这种情况下得到的是近似值。

13015 计算机系统原理【第二章考点解析】

考点5 二、八、十六进制数的相互转换

1）**八进制**数和**二进制**数之间的转换：**解题思路：一拆三**

八进制数字与二进制数的对应关系如下

$(0)_8=000$　$(1)_8=001$　$(2)_8=010$　$(3)_8=011$

$(4)_8=100$　$(5)_8=101$　$(6)_8=110$　$(7)_8=111$

例 2.8　将$(13.724)_8$转换成二进制数。

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

$(\sout{001}\ 011.111\ 010\ 1\sout{00})_2$

2）**十六进制**数和**二进制**数之间的转换：**解题思路：一拆四**

例 2.9　将十六进制数$(2B.5E)_{16}$转换成二进制数。

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

$(\sout{00}10\ 1011.0101\ 111\sout{0})_2$

13015 计算机系统原理【第二章考点解析】

考点5 二、八、十六进制数的相互转换

3）**二进制**数和**八进制**数之间的转换：**解题思路：三合一**

注意：小数点“.”，

① ←. 左边，由低位开始合，不够三位，高位补0

② .→ 右边，由高位开始合，不够三位，低位补0

例 2.8　将$(13.724)_8$转换成二进制数。

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

$(\sout{00}1\ 011.111\ 010\ 1\sout{00})_2$

4）**二进制**数和**十六进制**数之间的转换：**解题思路：四合一**

例 2.9　将十六进制数$(2B.5E)_{16}$转换成二进制数。

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

$(\sout{00}10\ 1011.0101\ 111\sout{0})_2$

13015 计算机系统原理【第二章考点解析】

考点5 二、八、十六进制数的相互转换

24. 简述计算机内部和外部需要进行数制转换的原因。

【答案】：【2024年10月】

计算机内部所有信息都采用二进制编码表示。但在计算机外部大都采用八、十或十六进制表示形式。因此，计算机在数据输入后或输出前都必须实现这些进制数和二进制数之间的转换。

13015 计算机系统原理【第二章考点解析】

考点6 定点数的编码表示

定点数编码表示方法主要有以下4种：

1）**原码**

2）**补码**

3）**反码**

4）**移码**：等于**补码符号位**取反

通常将**数值数据**在计算机内部编码表示的数称为**机器数**。

比如：0000 0101

机器数真正的值（即现实世界中带有正负号的数）称为机器数的**真值**。

比如：5

13015 计算机系统原理【第二章考点解析】

考点7 原码和反码表示法

1）**原码**：“**符号-数值**”表示法（**0：正**；**1：负**）

比如：0000 0101

原码 0 有两种表示形式：

$$[+0]_{原}=0\ 00\cdots0$$
$$[-0]_{原}=1\ 00\cdots0$$

对于真值 **-10** （**-1010 B**），用 8 位原码表示的机器数为 **1000 1010 B**（8AH 或 0x8A）

对于真值 **-0.625** (**-0.101 B**)，用 8 位原码表示的机器数为 **1101 0000 B**（D0H 或 0xD0）

2）**反码**：**1 → 0; 0 → 1**

13015 计算机系统原理【第二章考点解析】

考点7 原码和反码表示法

4. 负零的原码表示为

A. 1 00 …0　　B. 0 00 …0　　C. 1 11 …1　　D. 0 11 …1

【答案】：A【2025年10月】

【解析】：

原码中，最高位为符号位（0 正，1 负），其余为数值。

负零：符号位 1，数值为 0 → 符号位 1 加 全零数值部分 → 1000...0

（假设 8 位即 1000 0000）

原码 0 有两种表示形式：$[+0]_{原}=0\ 00\cdots0$

$[-0]_{原}=1\ 00\cdots0$

13015 计算机系统原理【第二章考点解析】

考点8 补码表示法

补码：实现**加减法统一**，即用**加法**来实现**减法**运算。

（1）模运算

“钟表” 是一个典型的模运算系统，其模数为 12。

假定现在钟表时针指向 **10** 点，要将它拨向 **6** 点，则有以下两种拨法。

1）倒拨 4 格：10−4=6

2）顺拨 8 格：10+8=18≡6(mod12)

所以在模 12 系统中，10**−4**≡10+(**12−4**)≡10+8(mod12)。即：**−4≡8(mod12)**。

例 2.10 假定在“钟表”上只能顺拨时针，则如何用顺拨的方式实现将 10 点倒拨 4 格，拨动后钟表上是几点？

解：“钟表”是一个模运算系统，其模为 12。因为 $10-4\equiv10+(12-4)\equiv10+8\equiv6(\mathrm{mod}\ 12)$，所以，可从 10 点顺拨 8 格来实现倒拨 4 格，最后拨到 6 点。

13015 计算机系统原理【第二章考点解析】

考点8 补码表示法

例 2.11 假定算盘只有 4 档，且只能做加法，则如何用该算盘计算 9828−1928 的结果？

解： 这个算盘是一个“4 位十进制数”模运算系统，其模为 10^4。

$$9828-1928 \equiv 9828+(10^4-1928) \equiv 9828+8072 \equiv 7900 \pmod{10^4}$$

可用 9828 加 8072（−1928 的补码）来实现 9828 减 1928 的功能。

9828+8072=17900 → $9828+8072 \equiv 7900$

以上发生了“**溢出**”现象

13015 计算机系统原理【第二章考点解析】

考点8 补码表示法

2. 计算机系统中采用补码运算的目的是

A. 与手工运算方式保持一致　　B. 提高运算速度

C. 提高运算的精度　　D. 简化计算机的设计

【答案】：D【2025年04月】

【解析】：

补码：实现**加减法统一**，即用**加法**来实现**减法**运算。

补码允许加法和减法使用相同的硬件电路（减法变加法，无需单独减法电路）

13015 计算机系统原理【第二章考点解析】

考点9 真值求补码

所以在模 12 系统中，10−4≡10+(12−4)≡10+8(mod12)。即：**−4≡8(mod12)**。

正数的补码是它本身；**负数**的补码等于模与该负数**绝对值**之差。

1）当 X_T 为正数时，$[X_T]_{补}=X_T=M+X_T(\bmod M)$。

2）当 X_T 为负数时，$[X_T]_{补}=M-|X_T|=M+X_T(\bmod M)$。

例 2.12 分别求出补码的位数为 n 和 $n+1$ 时“-2^{n-1}”的补码表示。

解： 当补码的位数为 n 位时，其模为 2^n。

$$[-2^{n-1}]_{补}=2^n-2^{n-1}=2^{n-1}=10\cdots0\ (n-1\text{ 个 }0)(\bmod 2^n)$$

当补码的位数为 $n+1$ 位时，其模为 2^{n+1}。

$$[-2^{n-1}]_{补}=2^{n+1}-2^{n-1}=2^n+2^{n-1}=110\cdots0\ (n-1\text{ 个 }0)(\bmod 2^{n+1})$$

例 2.13 假设补码的位数为 n，求“−1”的补码表示。

解： 根据补码定义，有：$[-1]_{补}=2^n-1=11\cdots1(n\text{ 个 }1)$。

例 2.14 求 0 的补码表示。

解： 根据补码的定义，有：$[+0]_{补}=[-0]_{补}=2^n\pm0=100\cdots0=00\cdots0(\bmod 2^n)$。

13015 计算机系统原理【第二章考点解析】

考点9 真值求补码

例 2.15　假设补码的位数为 8，求 110 1100 和−110 1100 的补码表示。

解：补码的位数为 8，说明补码数值部分有 7 位，故

$$[110\ 1100]_{补}=2^8+110\ 1100=1\ 0000\ 0000+110\ 1100=0110\ 1100(\mathrm{mod}\ 2^8)$$

$$[-110\ 1100]_{补}=2^8-110\ 1100=1\ 0000\ 0000-110\ 1100$$

13015 计算机系统原理【第二章考点解析】

考点9 真值求补码

真值求补码的**简便方法**为：

1）若：**正数**，则**符号位**取**0**，**数值部分**：**不变**；

2）若：**负数**，则**符号位**取**1**，**数值部分**：**各位取反，末位加1**

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

	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

13015 计算机系统原理【第二章考点解析】

考点9 真值求补码

14. 负数的补码等于_____与该负数绝对值之_____。

【答案】：模，差【2025年10月】

【解析】：**正数**的补码是它本身；**负数**的补码等于模与该负数**绝对值**之差。

所以在模 12 系统中，$10-4\equiv10+(12-4)\equiv10+8(\mathrm{mod}12)$。即：$-4\equiv8(\mathrm{mod}12)$。

13015 计算机系统原理【第二章考点解析】

考点10 补码求真值

补码求真值的**简便方法**为：

1）若**符号位**为**0**，则**真值**的**符号**为**正数**，**数值部分**：**不变**；

2）若**符号位**为**1**，则**真值**的**符号**为**负数**，**数值部分**：**各位取反，末位加1**

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

解：$X_T=-(100\ 1011+1)=-100\ 1100$

	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

13015 计算机系统原理【第二章考点解析】

考点11 补码求原码相反数的补码

解题思路：**符号位**+数值位：**各位取反，末位加1**

备注：与【考点9 真值求补码】和【考点10 补码求真值】比较，只有【数值位】**各位取反，末位加1**

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

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

	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

例 2.19　已知$[X_T]_{补}=1\ 000\ 0000$，求$[-X_T]_{补}$

解：$[-X_T]_{补}=0\ 111\ 1111+0\ 000\ 0001=1\ 000\ 0000$（结果溢出）

13015 计算机系统原理【第二章考点解析】

考点12 二进制整数

二进制整数分为以下两种：

1）无符号整数（默认数的符号为正）

n 位无符号整数可表示的数的范围为 0 ~ (2^n-1)

例如，8 位无符号整数的表示范围为 0 ~ (2^8-1) → 0 ~ 255

2）带符号整数（用补码来表示）

n 位带符号整数可表示的数的范围为 -2^{n-1} - $(2^{n-1}-1)$

例如，8 位带符号整数的表示范围为 -2^7 ~ (2^7-1) → -128 ~ +127

例 2.19 已知$[X_T]_{补}=1\ 000\ 0000$，求$[-X_T]_{补}$

解： $[-X_T]_{补}=0\ 111\ 1111+0\ 000\ 0001=1\ 000\ 0000$（结果溢出）

13015 计算机系统原理【第二章考点解析】

考点12 二进制整数

15. 在两个同号数相加时，当相加得到的_____超出了 n 位数可表示的范围时出现这种情况，此时发生了_____现象。

【答案】：和(或结果)　溢出【2024年10月】

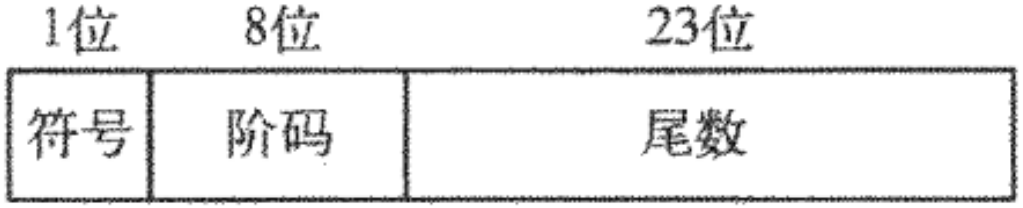
13015 计算机系统原理【第二章考点解析】

考点13 IEEE754浮点数标准

在这个标准中，提供了两种基本浮点格式：

1）**32位单精度**（**1**+**8**+**23**）

2）64位双精度格式（**1**+**11**+**52**）



解题思路：（**以32位单精度为准**）

1）先转为二进制　　【**指数**助记：左移正，右移负】

2）转为二进制的**科学计数法**，格式为：$1.F \times 2^n$，**1011.011** → **1.011011×2^3**

3）确定符号(**1**位)：**0：正**；**1：负**

4）确定阶码(**8**位)：**n**+**127**[**偏置常数**=$2^{n-1}-1=2^7-1$]，转为**二进制数**　　【阶用**移码**表示】

5）确定尾数(**23**位)：F，后面补0，补够**23**位

6）结合 **3）4）5）步骤** 得出IEEE754单精度浮点数的机器数

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13 IEEE754浮点数标准

例 2.21　将十进制数−0.75 转换为 IEEE 754 的单精度浮点数格式表示。

解题思路：

1）先转为二进制：$(-0.75)_{10} = (-0.11)_2$

2）转为二进制的**科学计数法**，-0.11 = -1.1 × 2^{-1}次方

3）确定符号(**1**位)：**1**（**0：正**；**1：负**）

4）确定阶码(**8**位)：**-1**+**127**=126，转为二进制为：**0111 1110**

5）确定尾数(**23**位)：10000000000000000000000

6）结合 **3）4）5）步骤** 得出IEEE754单精度浮点数的机器数：

10111111010000000000000000000000

7）四合一，转为**十六进制**：**BF40 0000H**

13015 计算机系统原理【第二章考点解析】

考点14 机器数的IEEE754浮点数转真值

【考点13 IEEE754浮点数标准】逆运算

解题思路：

1）先转为二进制，比如：**11000000101000000000000000000000**（32位）

2）确定符号：**0：正**；**1：负**

3）确定阶码：8位二进制**阶码转换为十进制**，再**-127** 【**指数**】

4）确定尾数：23位二进制**尾数转换为十进制**，转换为 0.F 格式

5）结合 **2）3）4）步骤**得出真值：**1.F** $\times 2^{n}$

13015 计算机系统原理【第二章考点解析】

考点14 机器数的IEEE754浮点数转真值

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）先转为二进制：**11000000101000000000000000000000**

2）确定符号(1位)：**1**（**0：正**；**1：负**）

3）确定阶码(8位)：**10000001**，转为真值：**129**，**129-127**=2，所以是2^2

4）确定尾数(**23**位)：**01**00000000000000000000，小数部分：0.01，二进制转为十进制：**0.25**

5）结合 **2）3）4）步骤**得出：$-1.25 \times 2^2 = -5.0$

13015 计算机系统原理【第二章考点解析】

考点15 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**

1）**int** ：$-2^{31} \sim 2^{31}-1$（即 -2147483648 ~ 2147483647）

2）**float** ：$-3.4 \times 10^{38} \sim 3.4 \times 10^{38}$，**精度**约为 **6-7** 位有效数字

3）**double** ：$-1.7 \times 10^{308} \sim 1.7 \times 10^{308}$，**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**。

比如：int b = 123456789; float fb = b; fb = 123456792.000000 (舍入后精度丢失)

② **int → double** 或 **float → double**

因为double类型的有效位数更多，故能保留精确值。

比如：int a = 1234567890;double da = a; da = 1234567890.000000 (无舍入)

比如：float b = 123.456787;double db = b; db = 123.456787109375 (无新增舍入)

13015 计算机系统原理【第二章考点解析】

考点15 C语言中的浮点数类型

③ **double → float**

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

比如：double a = 1.234567890123456;　float fa = a;　fa = 1.2345679

④ **double → int　　或　　float → int**

因为int没有小数部分，所以可能会截断

比如：double d = 1.999;　　int a = d;　a = 1;

1.999被转换为1，因为int表示的范围更小，超出 int 范围则溢出，可能溢出

比如：float f = 1.999;　　int a = f;　a = 1;

1.999被转换为1，因为int表示的范围更小，超出 int 范围则溢出，可能溢出

总结：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**

13015 计算机系统原理【第二章考点解析】

考点15 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+∞、−∞ 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16 非数值数据的编码表示

非数值数据分为以下：

1）逻辑值

核心只有两个值：真(1) 和 假(0)，按位进行

2）西文字符

目前计算机中使用最广泛的西文字符集及其编码是 ASCII 码

每个字符由 7 个二进位，$b_6b_5b_4b_3b_2b_1b_0$表示，

例如：0110111 → 左边是高位，右边的低位

3）汉字字符

① 汉字的输入码，又称外码

② 字符集与汉字内码　　（ASCII 码）

③ 汉字的字模点阵码和轮廓描述

13015 计算机系统原理【第二章考点解析】

考点16 非数值数据的编码表示

11. 指令所处理的非数值数据主要包括_____数据和_____数据。

【答案】：逻辑 字符【2024年10月】

13015 计算机系统原理【第二章考点解析】

考点17 数据的宽度和单位

1）比特：也叫：位，用 **b** 表示

2）字节：一个字节等于 **8** 位，用 **B** 表示　　**1B = 8b**

3）字：一般情况下，可以理解为**双字节**，**16** 位

4）**字长**：CPU内部用于整数运算的数据通路的宽度

例如：平常所说的机器是 32 位还是 64 位，指的就是字长

存储容量单位如下：

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17 数据的宽度和单位

23. 简述字和字长概念。

【答案】：【2025年04月】

字用来表示被处理信息的单位，用于度量各种数据类型的宽度。

字长是指 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【第二章考点解析】

考点18 数据的存储和排列顺序

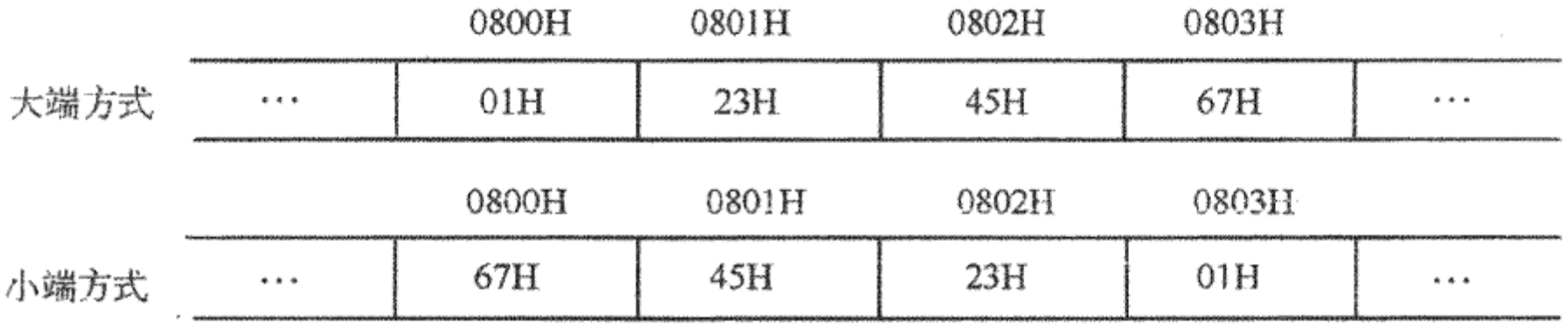


图 2.5 大端方式和小端方式

假定int型变量i的地址为0800H(低地址)， 地址是都从低-高排列，机器数为01234567H(左高右低)

01H 为MSB表示最高有效字节，67H 为LSB表示最低有效字节

1）大端方式：机器数顺序排列，01H、23H、45H、67H （“正着装”）

核心：数据的 “高位字节” 存在内存的 “低地址” 里。 助记：高大

2）小端方式：机器数逆序排列，67H、45H、23H、01H （“反着装”）

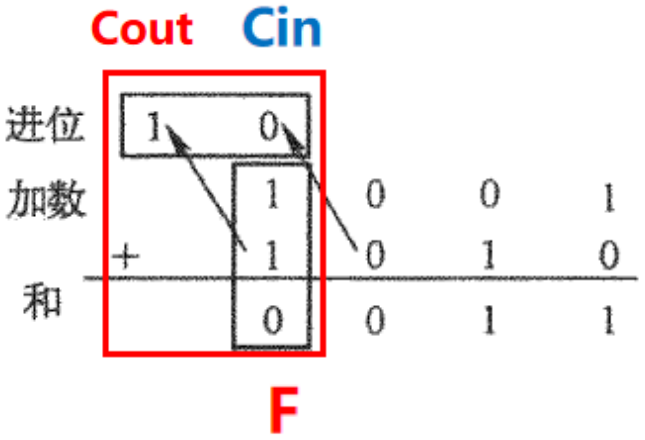
核心：数据的 “低位字节” 存在内存的 “低地址” 里。 助记：低小

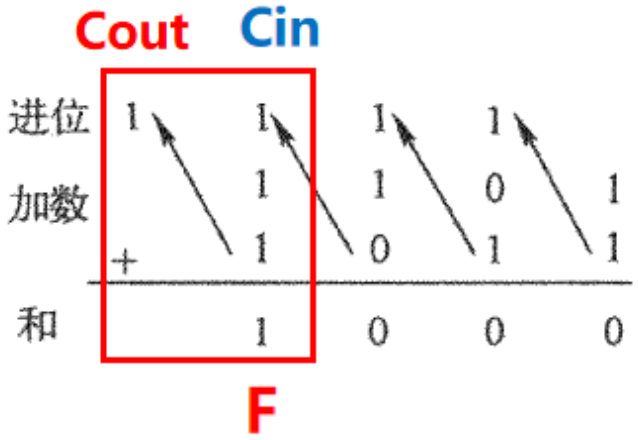
13015 计算机系统原理【第二章考点解析】

考点19 加法器和算术逻辑部件

全加器的**两个加数**为 **A** 和 **B**，低位进位为 **Cin**，

相加的和为 **F**，向高位的进位为 **Cout**。





A	B	Cin	F	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

图 2.6　全加器真值表

加法器实际上是**无符号数**加法器。

13015 计算机系统原理【第二章考点解析】

考点19 加法器和算术逻辑部件

24. 简述算术逻辑部件的工作原理。

【答案】：【2024年04月】

用来完成基本的逻辑运算和定点数加减运算，各类定点乘除运算和浮点数运算可利用加法器、ALU和移位器来实现，因此基本的运算部分是加法器、ALU和移位器，ALU的核心部件是加法器。

13015 计算机系统原理【第二章考点解析】

考点20 带标志加法器

带标志加法器

1）**溢出**标志：表示**有符号数**运算时，结果超出了当前位数能表示的范围。

$OF = C_n \oplus C_{n-1}$，其中 C_n 是**最高位**（符号位）的进位，C_{n-1} 是**次高位**的进位。

比如：以 4 位**有符号数**为例，范围是 **-8 ~ +7**

计算 **+6 + +3**

	0	1	1	0	(+6)
+	0	0	1	1	(+3)
	1	0	0	1	(−7) (结果错误，发生溢出)

① 次高位（第2位）向最高位（第3位）有进位 → C_{n-1} = 1

② 最高位（第3位）向更高位无进位 → C_n = 0

所以 $OF = C_n \oplus C_{n-1}$ = 1，表示运算溢出。

13015 计算机系统原理【第二章考点解析】

考点20 带标志加法器

带标志加法器

2）**符号**标志：表示**有符号数**运算结果的符号，直接等于结果的**最高位**（符号位）。

$SF = F_{n-1}$ ，是结果的最高位（**0：正**；**1：负**）。

比如：以 4 位**有符号数**为例，范围是 **-8 ~ +7**

计算 **+5 + -3**

	0	1	0	1	(+5)
+	1	1	0	1	(−3)
	0	0	1	0	(+2)

结果最高位是 0 → SF = 0，表示结果为**正**。

13015 计算机系统原理【第二章考点解析】

考点20 带标志加法器

带标志加法器

3）零标志：**运算结果**为 0 时，ZF = 1；否则 ZF = 0

计算 **+4 + -4**

	0	1	0	0	(+4)
+	1	1	0	0	(−4)
	0	0	0	0	0

结果是 0 → ZF = 1。

13015 计算机系统原理【第二章考点解析】

考点20 带标志加法器

带标志加法器

4）**进位/借位**标志：**无符号数**运算时的进位（加法）或借位（减法）标志。

当 Cin = 0（加法），CF = Cout（直接取最高位的进位）；

当 Cin = 1（减法，可看作加负数），CF = ¬Cout（对最高位的进位取反）

计算 1111 + 0001（4 位**无符号数**，范围 0 ~ 15）：

	1	1	1	1	15
+	0	0	0	1	1
	0	0	0	0	0

最高位有进位 → Cout = 1

因为 Cin = 0，所以 CF = Cout = 1，表示产生了进位。

13015 计算机系统原理【第二章考点解析】

考点21 补码加减运算器

公式：$[x+y]_{补}=[x]_{补}+[y]_{补}(\mod 2^n)$

$[x-y]_{补}=[x]_{补}+[-y]_{补}(\mod 2^n)$

例 2.24 用 4 位补码计算“−7−6”和“−3−5”的值。

$[-7]_{原}$=1111，**数值位**取反：1000，再加1，$[-7]_{补}$=1001，**其他的同理**

解： $[-7]_{补}=1001,[-6]_{补}=1010,[-3]_{补}=1101,[-5]_{补}=1011$

$[-7-6]_{补}=[-7]_{补}+[-6]_{补}=1001+1010=0011(+3)$

$[-3-5]_{补}=[-3]_{补}+[-5]_{补}=1101+1011=1000(-8)$

因为4 位补码的可表示范围为：**-8 ~ +7**，所以

① -7-6= -13<-8，所以，结果0011(+3)一定发生了溢出，是一个错误的值。

② -3-5= -8=-8，所以，结果1000(-8)，没有超出范围，因而没有发生溢出，是一个正确的值。

13015 计算机系统原理【第二章考点解析】

考点22 浮点数加减运算

浮点数加减运算过程中，需要经过以下步骤

1）对阶

2）尾数加减

3）规格化

4）舍入

5）溢出判断和溢出处理

13015 计算机系统原理【第二章考点解析】

考点22 浮点数加减运算

对阶的原则：**小阶向大阶**看齐，**阶小的那个数的尾数右移**，右移的位数等于两个阶的差的绝对值。

E_x、E_y分别是浮点数 x 和 y 的阶

$$[E_x - E_y]_{补} = [E_x]_{移} + [-[E_y]_{移}]_{补} (\mathrm{mod}\ 2^n)$$

例 2.29　若 x 和 y 为 float 变量，$x=1.5$，$y=-125.25$，请给出计算 $x+y$ 过程中的对阶结果。

解：x = 1.5 = 1.1B = 1.1 B × 2^0 B，机器数为 0 0111 1111 100 0000 0000 0000 0000 0000

y = -125. 25 = -111 1101. 01 B = -1.1111 0101 B × 2^6 B，

机器数为 1 **1000 0101** 111 1010 1000 0000 0000 0000

小阶向大阶看齐，所以：应对 x 的**尾数右移 6 位**，对阶后 x 的阶码为 **1000 0101**，

尾数为0.00 0001 100 0000…0000

13015 计算机系统原理【第二章考点解析】

考点22 浮点数加减运算

例 2.30　用 IEEE 754 单精度浮点数加减运算计算 0.5+(−0.4375)=？

$(0.5)_{10} = (0.1)_2 = 1.0\times\mathbf{2}^{-1}$，机器数：0 01111110 00000000000000000000000

$(-0.4375)_{10} = (0.0111)_2 = 1.11\times\mathbf{2}^{-2}$，机器数：1 01111101 11000000000000000000000

1）对阶

对阶的目的是让两个数的**指数相同**，这样尾数才能直接相加。

对阶的原则：**小阶向大阶**看齐，**阶小的那个数的尾数右移**，-2 → -1，尾数右移 1 位。

调整后 -0.4375 的尾数：原尾数是 1.11 → 右移 1 位 → 0.111【**备注**：**小数点前的1是隐藏的1**】

对阶后：

0.5 的尾数：1.000

-0.4375 的尾数：-0.111

13015 计算机系统原理【第二章考点解析】

考点22 浮点数加减运算

2）尾数加减

现在两个数的指数相同，可以直接把尾数相加，符号参与运算：1.000+(−0.111)=**0.001**

3）规格化

规格化要求尾数是 1.F 的形式（隐藏整数位 1）。

当前尾数是 **0.001**，需要左移 3 位变成 1.000，即：**0.001** = 1.000×$\mathbf{2}^{-3}$，

新指数=126（-1+127）-3=**123**，新的尾数（隐含 1 恢复形式）：**1.000**...0，尾数部分 23 位全 0。

4）舍入

我们的尾数在规格化后是 **1.000**，没有多余的位需要舍入，所以尾数保持 1.000。

5）溢出判断和溢出处理

指数范围：IEEE 754 单精度的指数范围是 -126 ~ +127。

我们得到的指数是 -3，在有效范围内，所以**没有溢出**。

13015 计算机系统原理【第二章考点解析】

考点22 浮点数加减运算

原来指数是 **126**（对齐后的指数=-1+127），新指数=126（-1+127）-3=**123**

最终结果

二进制表示：0 **01111011** 00000000000000000000000

转换为十进制：$1.0\times2^{126-127-3}=0.0625$，所以：0.5+(−0.4375)=0.0625

或

转换为十进制：$1.0\times2^{123-127}=0.0625$，所以：0.5+(−0.4375)=0.0625

注意：指数 = 阶码 - 127[偏置常数]

13015 计算机系统原理【第二章考点解析】

考点22 浮点数加减运算

15. 对阶时，使小阶向大阶看齐，使小阶的_____向右移位，移位的位数等于两个阶的_____的绝对值。

【答案】：尾数、差【2024年04月】

14. 浮点数加减运算过程中，需要经过_____、尾数加减、_____和舍入4个步骤。

【答案】：对阶、规格化【2024年10月】】

13. 浮点数加减运算过程中，需要经过对阶、_____加减、_____和舍入4个步骤。

【答案】：尾数、规格化【2025年04月】】

13015 计算机系统原理【第二章考点解析】

考点23 浮点数乘除运算

浮点数乘除运算过程中，需要经过以下步骤

1）尾数相乘、阶码相加

$0.123\times10^{5}\times0.456\times10^{2}$

$=0.123\times0.456\times10^{5+2}$

尾数相除、阶码相减

$0.123\times10^{5}\ /\ 0.456\times10^{2}$

$=0.123\ /\ 0.456\times10^{5-2}$

2）尾数规格化

3）尾数舍入处理

4）阶码溢出判断

13015 计算机系统原理【第二章考点解析】

考点23 浮点数乘除运算

4. 下列关于浮点运算器的描述，正确的是

A. 浮点运算器可用阶码部件和尾数部件来实现

B. 阶码部件可实现加、减、乘、除四种运算

C. 阶码部件可进行阶码相加、相减和相乘操作

D. 尾数部件只进行乘法和除法运算

【答案】：A【2024年10月】

【解析】：

浮点运算中，阶码表示范围，尾数表示精度

阶码部件主要进行阶码相加、减和比较操作

尾数部件能实现加、减、乘和除四种基本算术运算

谢 谢 大 家