Lecture 6-1: Data Representation 3

数据的机器级表示 3

海明校验码

- 由Richard Hamming于1950年提出，目前还被广泛使用。
- 主要用于存储器中数据存取校验。
- 基本思想：奇偶校验码对整个数据编码生成一位校验位。因此这种校验码检错能力差，并且没有纠错能力。如果将整个数据按某种规律分成若干组，对每组进行相应的奇偶检测，就能提供多位检错信息，从而对错误位置进行定位，并将其纠正。
  - 海明校验码实质上就是一种多重奇偶校验码。
- 处理过程：
  - 最终比较时按位进行异或，以确定是否有差错。
  - 这种异或操作所得到的结果称为故障字（syndrome word）。显然，校验码和故障字的位数是相同。

每一组一个校验位，校验码位数等于组数！

每一组内采用一位奇偶校验！

校验码位数的确定

- 假定数据位数为n，校验码为k位，则故障字位数也为k位。k位故障字所能表示的状态最多是2^K，每种状态可用来说明一种出错情况。
- 若只有一位错，则结果可能是：
  - 数据中某一位错 (n种可能)
  - 校验码中有一位错 (k种可能)
  - 无错 (1 种可能)

  } 1+n+k种情况

假定最多有一位错，则n和k必须满足下列关系：

$$2^K \geqslant 1+n+k, \quad \text{即：} 2^K-1 \geqslant n+k$$

- 有效数据位数和校验码位数间的关系
- 从表中可看出，当数据有8位时，校验码和故障字都应有4位。

说明：4位故障字最多可表示16种状态，而单个位出错情况最多只有12种可能（8个数据位和4个校验位），再加上无错的情况，一共有13种。所以，用16种状态表示13种情况应是足够了。

有效数据位数和校验码位数间的关系

	单纠错		单纠错/双检错	
数据位数	校验位数	增加率	校验位数	增加率
8	4	50	5	62.5
16	5	31.25	6	37.5
32	6	18.75	7	21.875
64	7	10.94	8	12.5
128	8	6.25	9	7.03
256	9	3.52	10	3.91

海明码的分组

- **基本思想：** 数据位和校验位按某种方式排列为一个（n+k）位的码字，将该字中每个出错位的位置与故障字的数值建立关系，通过故障字的值确定该码字中哪一位发生了错误，并将其取反来纠正。

  根据上述基本思想，按以下规则来解释各故障字的值。

  规则1： 若故障字每位全部是0，则表示没有发生错误。

  规则2：若故障字中有且仅有一位为1，则表示校验位中有一位出错，因而不需纠正。

  规则3：若故障字中多位为1，则表示有一个数据位出错，其在码字中的出错位置由故障字的数值来确定。纠正时只要将出错位取反即可。

海明码的分组

以8位数据进行单个位检错和纠错为例说明。

假定8位数据M= $M_8M_7M_6M_5M_4M_3M_2M_1$，4位校验位P=$P_4P_3P_2P_1$。根据规则将M和P按一定的规律排到一个12位码字中。

据规则1，故障字为0000时，表示无错。

据规则2，故障字中有且仅有一位为1时，表示校验位中有一位出错。此时，故障字只可能是0001、0010、0100、1000，将这四种状态分别代表校验位中第P_1、P_2、P_3、P_4位发生错误，因此，它们分别位于码字的第1、2、4、8位。

据规则3，将其他多位为1的故障字依次表示数据位M_1~M_8发生错误的情况。因此，数据位M_1~M_8分别位于码字的第0011(3)、0101(5)、0110(6)、0111(7)、1001(9)、1010(10)、1011(11)、

列为： $M_8M_7M_6M_5P_4M_4M_3M_2P_3M_1P_2P_1$

是逻辑顺序，物理上M和P是分开的！

这样，得到故障字S=$S_4S_3S_2S_1$的各个状态和出错情况的对应关系表，可根据这种对应关系对整个数据进行分组。

海明校验码分组情况

故障字$S_4S_3S_2S_1$每一位的值反映所在组的奇偶性

码字：$M_8M_7M_6M_5P_4M_4M_3M_2P_3M_1P_2P_1$

序号 / 含义 / 分组	1	2	3	4	5	6	7	8	9	10	11	12	故障字	正确	出错位
	P_1	P_2	M_1	P_3	M_2	M_3	M_4	P_4	M_5	M_6	M_7	M_8			1 2 3 4 5 6 7 8 9 10 11 12
第4组								√	√	√	√	√	S_4	0	0 0 0 0 0 0 0 1 1 1 1
第3组				√	√	√	√					√	S_3	0	0 0 0 1 1 1 1 0 0 0 1
第2组		√	√			√	√			√	√		S_2	0	0 1 1 0 0 1 1 0 0 1 1 0
第1组	√		√		√		√		√		√		S_1	0	1 0 1 0 1 0 1 0 1 0 1 0

数据位或校验位出错一定会影响所在组的奇偶性。

例：若M1出错，则故障字为0011，因而会改变S_2和S_1所在分组的奇偶性。故M1同时被分到第2(S_2)组和第1(S_1) 组。

问题：若P_1出错，则如何？若M_8出错，则如何？

P_1~0001，分在第1组；M_8~1100，分在第4组和第3组

校验位的生成和检错、纠错

- 分组完成后，就可对每组采用相应的奇（偶）校验，以得到相应的一个校验位。
- 假定采用偶校验（取校验位Pi，使对应组中有偶数个1），则得到校验位与数据位之间存在如下关系：

$$P_1 = M_1 \oplus M_2 \oplus M_4 \oplus M_5 \oplus M_7$$

$$P_2 = M_1 \oplus M_3 \oplus M_4 \oplus M_6 \oplus M_7$$

$$P_3 = M_2 \oplus M_3 \oplus M_4 \oplus M_8$$

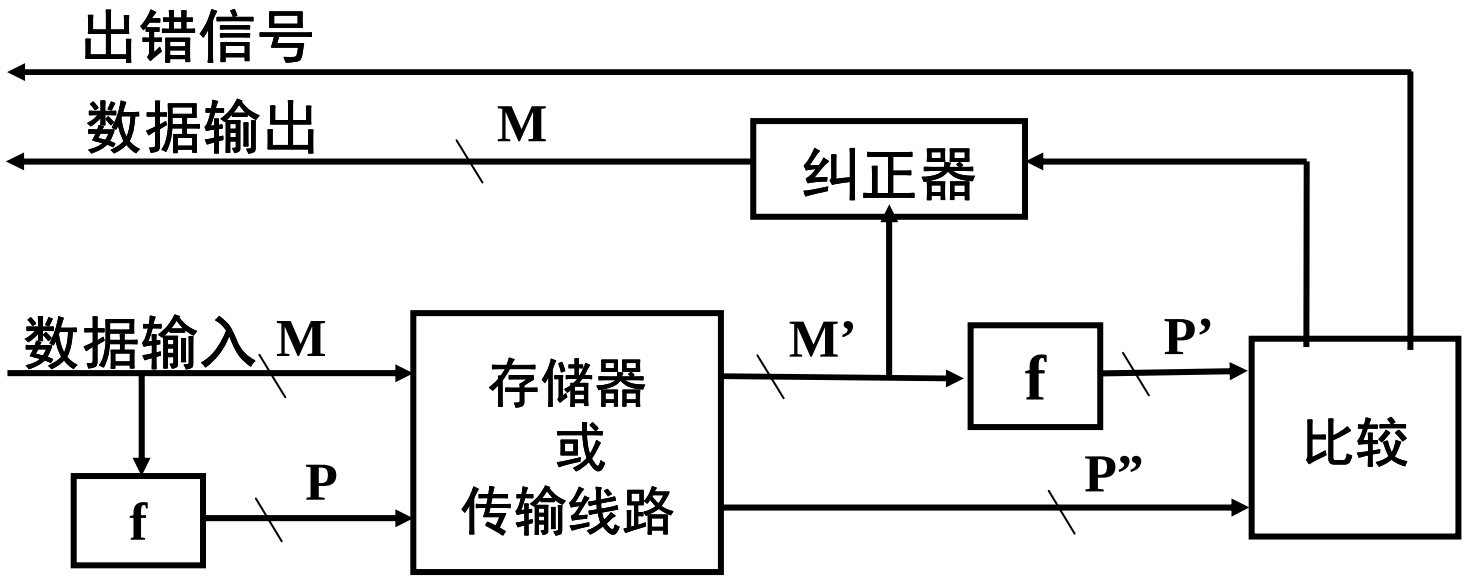
$$P_4 = M_5 \oplus M_6 \oplus M_7 \oplus M_8$$

海明校验过程

- 根据公式求出每一组对应的校验位Pi (i=1,2,3,4)
- 数据M和校验位P一起被存储，根据读出数据M’得新校验位P’
- 读出校验位P’’与新校验位P’ 按位进行异或操作，得故障字

$$S = S_4S_3S_2S_1$$

- 根据S的值确定：无错、仅校验位错、某个数据位错



海明码举例

假定一个8位数据M为：$M_8M_7M_6M_5M_4M_3M_2M_1$= 01101010，根据上述公式求出相应的校验位为：

$P_1 = M_1 \oplus M_2 \oplus M_4 \oplus M_5 \oplus M_7 = 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 = 1$

$P_2 = M_1 \oplus M_3 \oplus M_4 \oplus M_6 \oplus M_7 = 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 1$

$P_3 = M_2 \oplus M_3 \oplus M_4 \oplus M_8 = 1 \oplus 0 \oplus 1 \oplus 0 = 0$

$P_4 = M_5 \oplus M_6 \oplus M_7 \oplus M_8 = 0 \oplus 1 \oplus 1 \oplus 0 = 0$

假定12位码字 ($M_8M_7M_6M_5P_4M_4M_3M_2P_3M_1P_2P_1$) 读出后为：

(1) 数据位M’=M=01101010，校验位P’’=P=0011

(2) 数据位M’= 01111010，校验位P’’=P=0011

(3) 数据位M’=M=01101010，校验位P’’= 1011

要求分别考察每种情况的故障字。

(1) 数据位M’=M=01101010，校验位P’’=P=0011，即无错。

因为M’=M，所以P’= P， 因此 S = P’’⊕P’=P⊕P=0000。

海明码举例

(2) 数据位M'= 01111010，校验位P''=P=0011，即M_5错。

对M'生成新的校验位P'为：

$P_1' = M_1' \oplus M_2' \oplus M_4' \oplus M_5' \oplus M_7' = 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 0$

$P_2' = M_1' \oplus M_3' \oplus M_4' \oplus M_6' \oplus M_7' = 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 1$

$P_3' = M_2' \oplus M_3' \oplus M_4' \oplus M_8' = 1 \oplus 0 \oplus 1 \oplus 0 = 0$

$P_4' = M_5' \oplus M_6' \oplus M_7' \oplus M_8' = 1 \oplus 1 \oplus 1 \oplus 0 = 1$

故障字S为：

$S_1 = P_1' \oplus P_1'' = 0 \oplus 1 = 1$

$S_2 = P_2' \oplus P_2'' = 1 \oplus 1 = 0$

$S_3 = P_3' \oplus P_3'' = 0 \oplus 0 = 0$

$S_4 = P_4' \oplus P_4'' = 1 \oplus 0 = 1$

因此，错误位是第9位，排列的是数据位M_5，所以检错正确，纠错时，只要将码字的第9位（M_5）取反即可。

海明码举例

(3) 数据位M’=M=01101010，校验位P’’= 1011，

即：校验码第4位(P_4)错。

因为M’=M，所以P’= P，因此故障位S为：

$S_1= P_1’ \oplus P_1’’= 1 \oplus 1=0$

$S_2= P_2’ \oplus P_2’’=1 \oplus 1=0$

$S_3= P_3’ \oplus P_3’’=0 \oplus 0=0$

$S_4= P_4’ \oplus P_4’’=0 \oplus 1=1$

错误位是第1000位(即第8位)，这位上排列的是校验位P_4，所以检错时发现数据正确，不需纠错。

单纠错和双检错码

- 单纠错码（SEC）
  - 问题：上述(n=8/k=4)海明码的码距是几？
  - 码距d=3。因为，若有一位出错，则因该位至少要参与两组校验位的生成，因而至少引起两个校验位的不同。两个校验位加一个数据位等于3。

    例如，若M_1出错，则故障字为0011，即P_2和P_1两个校验位发生改变，12位码字中有三位（M_1、P_2和P_1）不同。
  - 根据码距与检错、纠错能力的关系，知：这种码制能发现两位错，或对单个位出错进行定位和纠错。这种码称为单纠错码（SEC）。

单纠错和双检错码

- 单纠错和双检错码（SEC-DED）
  - 具有发现两位错和纠正一位错的能力，则称为单纠错和双检错码（SEC-DED）。
  - 若要成为SEC-DED，则码距需扩大到d= 4。为此，还需增加一位校验位P_5，将P_5排列在码字的最前面，即：$P_5M_8M_7M_6M_5P_4M_4M_3M_2P_3M_1P_2P_1$，并使得数据中的每一位都参与三个校验位的生成。从表中可看出除了M_4和M_7外，其余位都只参与了两个校验位的生成。因此P_5按下式求值：

$$P_5= M_1 \oplus M_2 \oplus M_3 \oplus M_5 \oplus M_6 \oplus M_8$$

当任意一个数据位发生错误时，必将引起三个校验位发生变化，所以码距为4。

循环冗余码

循环冗余校验码（Cyclic Redundancy Check），简称CRC码

- 具很强的检错、纠错能力。
- 用于大批量数据存储和传送(如：外存和通信)中的数据校验。

  为什么大批量数据不用奇偶校验？

  在每个字符后增加一位校验位会增加大量的额外开销；尤其在网络通信中，对传输的二进制比特流没有必要再分解成一个个字符，因而无法采用奇偶校验码。
- 通过某种数学运算来建立数据和校验位之间的约定关系。

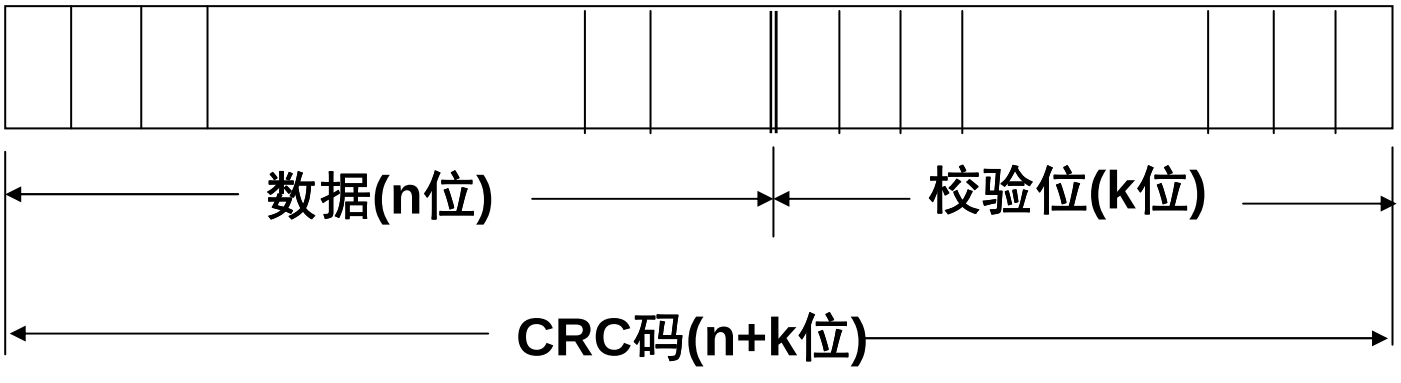
  奇偶校验码和海明校验码都是以奇偶检测为手段的。

网络或通信课程中会学到。

CRC码的检错方法

基本思想：

- 数据信息M(x)为一个n位的二进制数据，将M(x)左移k位后，用一个约定的“生成多项式”G(x)相除，G(x)是一个k+1位的二进制数，相除后得到的k位余数就是校验位。校验位拼接到M(x)后，形成一个n+k位的代码，称该代码为循环冗余校验（CRC）码，也称（n+k,n）码。
- 一个CRC码一定能被生成多项式整除，当数据和校验位一起送到接受端后，只要将接受到的数据和校验位用同样的生成多项式相除，如果正好除尽，表明没有发生错误；若除不尽，则表明某些数据位发生了错误。通常要求重传一次。



循环冗余码举例

校验位的生成：用一个例子来说明校验位的生成过程。

- 假设要传送的数据信息为：100011，即报文多项式为：

  $M(x)= x^5 + x + 1$。数据信息位数n=6。

- 若约定的生成多项式为：$G(x)= x^3+ 1$，则生成多项式位数为4位，所以校验位位数k=3，除数为1001。

- 生成校验位时，用$x^3 \cdot M(x)$去除以G(x)，即：100011000÷1001。

- 相除时采用“模２运算”的多项式除法。

循环冗余码举例

$X^3 \cdot M(x) \div G(x) = (x^8 + x^4 + x^3) \div (x^3 + 1)$

```
        100111
    ------------
1001 | 100011000
       1001
       ----
        0011
        0000
        ----
         0111
         0000
         ----
          1110
          1001
          ----
           1110
           1001
           ----
            1110
            1001
            ----
             111
```

(模２运算不考虑加法进位和减法借位，上商的原则是当部分余数首位是１时商取１，反之商取０。然后按模２相减原则求得最高位后面几位的余数。这样当被除数逐步除完时，最后的余数位数比除数少一位。这样得到的余数就是校验位，此例中最终的余数有3位。)

校验位为111，CRC码为100011 111。如果要校验CRC码，可将CRC码用同一个多项式相除，若余数为0，则说明无错；否则说明有错。例如，若在接收方的CRC码也为100011 111时，用同一个多项式相除后余数为0。若接收方CRC码不为100011 111时，余数则不为0。

小结

- 非数值数据的表示
  - 逻辑数据用来表示真/假或N位位串，按位运算
  - 西文字符：用ASCII码表示
  - 汉字：汉字输入码、汉字内码、汉字字模码
- 数据的宽度
  - 位、字节、字（不一定等于字长），k/K/M/G/…有不同的含义
- 数据的存储排列
  - 大端方式：用MSB存放的地址表示数据的地址
  - 小端方式：用LSB存放的地址表示数据的地址
  - 按边界对齐可减少访存次数
- 数据的纠错和检错
  - 奇偶校验：适应于一字节长数据的校验，如内存
  - 海明校验：各组内用奇偶校验，用于内存储器数据的校验
  - 循环冗余校验：用在通信和外存中，适合于大批量数据校验

附录： Decimal / Binary（十 / 二进制数）

- **The decimal number 5836.47 in powers of 10:**

$$5 \times 10^3 + 8 \times 10^2 + 3 \times 10^1 + 6 \times 10^0 + 4 \times 10^{-1} + 7 \times 10^{-2}$$

- **The binary number 11001 in powers of 2 :**

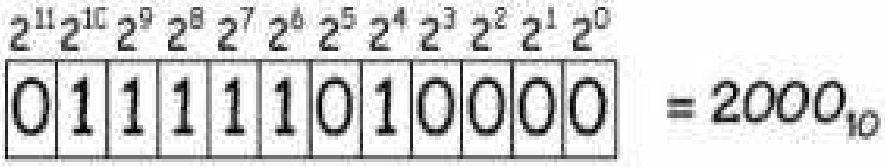
$$1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$= 16 + 8 + 0 + 0 + 1 = 25$$

- 用一个下标表示数的基（ **radix / base**）

$$11001_2 = 25_{10}$$

附录： Octal / Hexadecimal (八 / 十六进制数)



$$v = \sum_{i=0}^{n-1} 2^i b_i$$

2^3=8

03720

Octal - base 8

000 - 0
001 - 1
010 - 2
011 - 3
100 - 4
101 - 5
110 - 6
111 - 7

2^4=16

0x7d0

Hexadecimal - base 16

0000 - 0	1000 - 8
0001 - 1	1001 - 9
0010 - 2	1010 - a
0011 - 3	1011 - b
0100 - 4	1100 - c
0101 - 5	1101 - d
0110 - 6	1110 - e
0111 - 7	1111 - f

计算机用二进制表示所有信息！

为什么要引入 8 / 16进制？

8 / 16进制是二进制的简便表示。便于阅读和书写！

它们之间对应简单，转换容易。

在机器内部用二进制，在屏幕或其他外部设备上表示时，转换为8/16进制数，可缩短长度

附录： Conversions of numbers

(1) R进制数 => 十进制数

按“权”展开 **(a power of R)**

例1: $(10101.01)_2=1x2^4+1x\ 2^2+1x2^0+1x2^{-2}=(21.25)_{10}$

例2: $(307.6)_8=3x8^2+7x8^0+6x8^{-1}=(199.75)_{10}$

例1: $(3A.\ 1)_{16}=3x16^1+10x16^0+1x16^{-1}=(58.0625)_{10}$

(2)十进制数 => R进制数

整数部分和小数部分分别转换

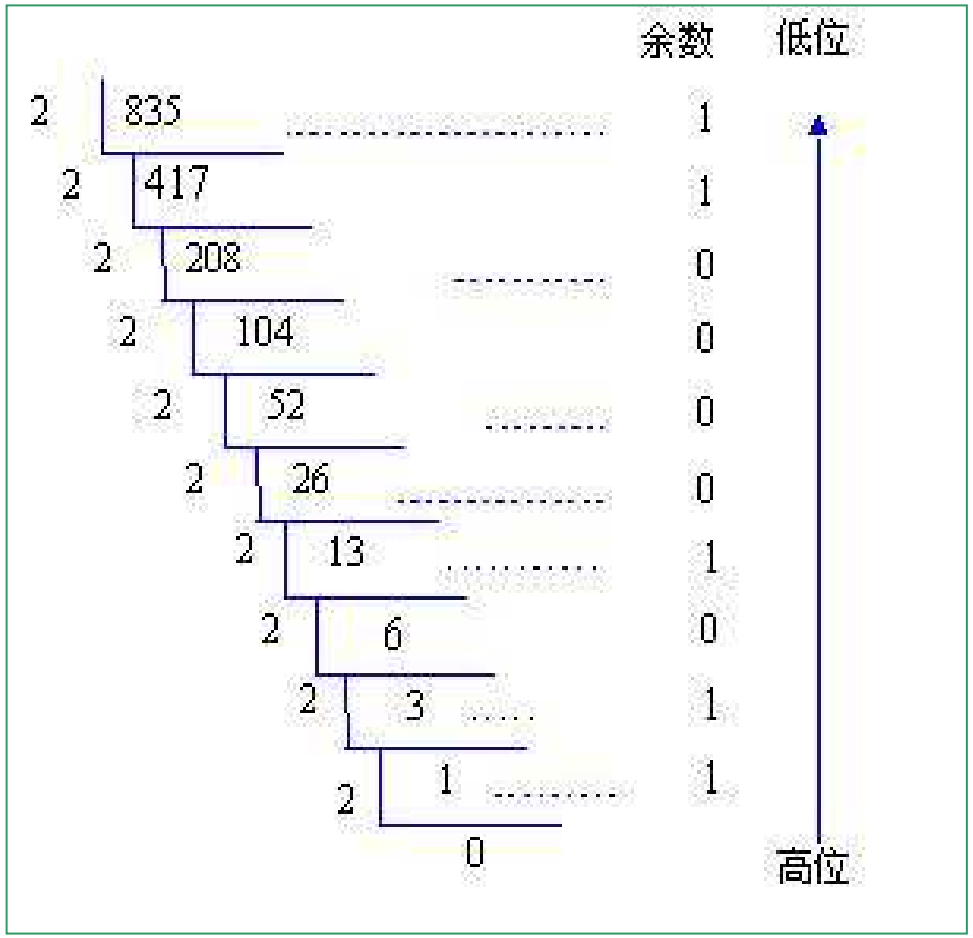
① 整数**(integral part)----**“除基取余，上右下左”

② 小数**(fractional part)----**“乘基取整，上左下右”

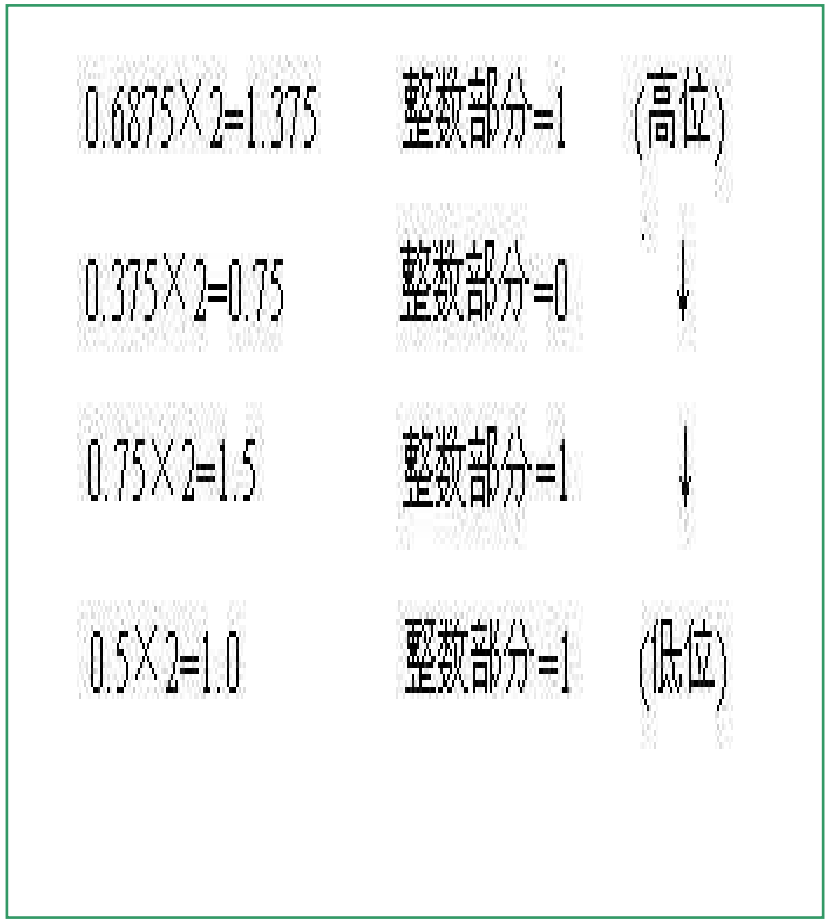
附录： Decimal to Binary Conversions

例1:$(835.6785)_{10}=(1101000011.1011)_2$

整数----“除基取余，上右下左”



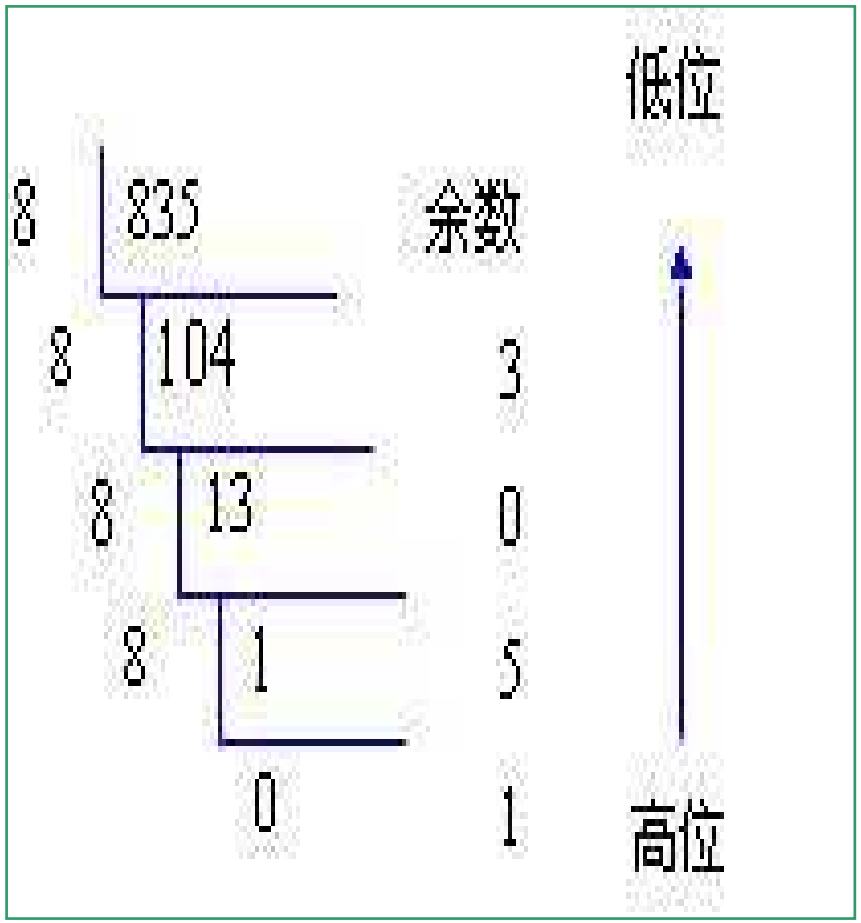
小数----“乘基取整，上左下右”



附录： Decimal to Binary Conversions

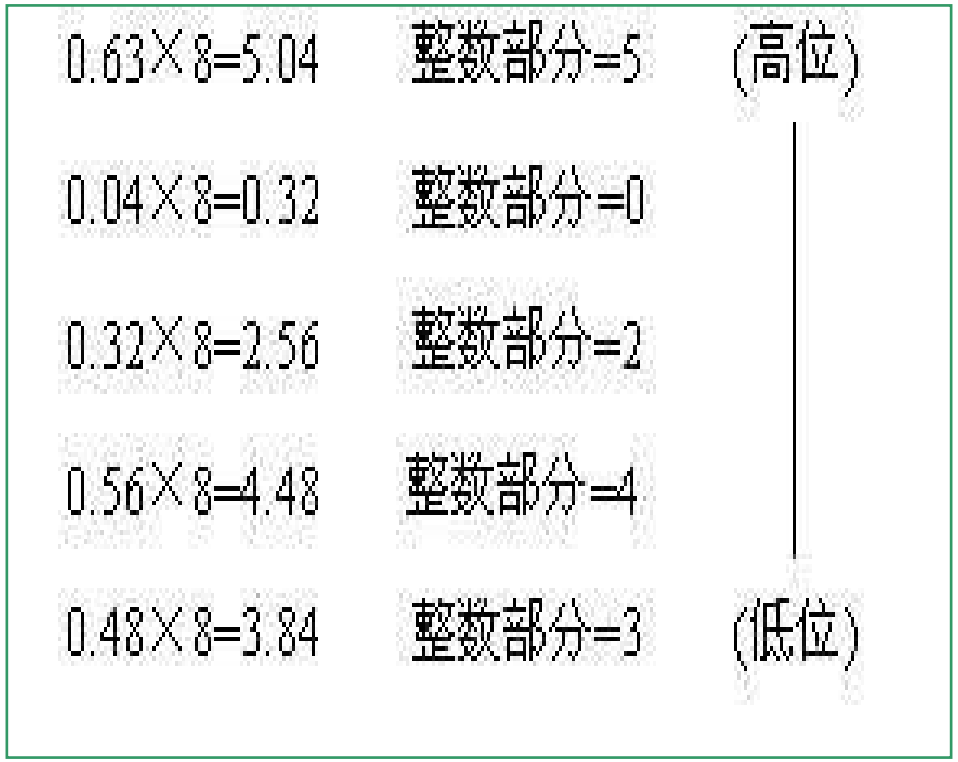
例2:$(835.63)_{10}=(1503.50243\ldots)_8$

整数----“除基取余，上右下左”



小数----“乘基取整，上左下右”

有可能乘积的小数部分总得不到0 ，此时得到一个近似值。



附录： Conversions of numbers

(3) 二/八/十六进制数的相互转换

① 八进制数转换成二进制数

$(13.724)_8=(\ 001\ 011\ .\ 111\ \ 010\ \ 100\ )_2=(1011.1110101)_2$

② 十六进制数转换成二进制数

$(2B.5E)_{16} = (00101011\ .\ 01011110)_2 = (101011.0101111)_2$

③ 二进制数转换成八进制数

$(0.10101)_2 = (\ 000\ .\ 101\ \ 010\ )_2 = (\ 0.52)_8$

④ 二进制数转换成十六进制数

$(11001.11)_2 = (\ 0001\ \ 1001\ .\ 1100\ )_2 = (\ 19.C\ )_{16}$琇菖

作业 一

习题1 课本 page 23

问题 6

9月27号交本子

习题2 课本 page 68

问题 7：(R1)=0000 017AH, (R2)=FFFF F895H

问题 13：变量的值为6144

问题 18：采用偶校验，接收方收到的校验位为1010

问题 20：需传送数据和接受到的数据分别是 110010、110011