北京高教精品教材项目

北京市高等教育精品教材立项项目

微机原理与接口技术教程

王克义　鲁守智　蔡建新　王文保　编著

高等院校计算机专业及专业基础课系列教材

北京大学出版社

北京市高等教育精品教材立项项目

微机原理与接口技术教程

王克义　鲁守智
蔡建新　王文保　编著

北京大学出版社
PEKING UNIVERSITY PRESS

内容简介

本书全面、系统地介绍了现代微型计算机的基本组成结构、工作原理及典型接口技术。主要内容包括：计算机科学技术基础，计算机/微型计算机的组成与结构，微处理器结构，指令系统与汇编语言程序设计，存储器及其接口，输入/输出及DMA接口，中断系统，串并行通信及其接口电路，模拟接口，总线技术，80x86/Pentium保护模式的软件体系结构，高性能微处理器及其相关技术等，最后还简单介绍了几种常见的计算机外部设备。

本书内容丰富，实用性强，特别注重基本概念和基本原理的讲解，并努力反映和吸收当前计算机发展的最新成果和技术。本书的各章节经过仔细的编排，内容由浅入深，循序渐进，逻辑性强，适合于自学，可作为高等学校计算机专业及理工科非计算机专业的计算机基础课程教材，也可作为高等教育自学考试及各类职业技术学校的教材或教学参考书。

图书在版编目(CIP)数据

微机原理与接口技术教程/王克义等编著. —北京：北京大学出版社，2004.12
(高等院校计算机专业及专业基础课系列教材)
ISBN 7-301-06367-9

Ⅰ. 微… Ⅱ. 王… Ⅲ. ①微机—理论—高等学校—教材 ②微型计算机—接口设备—高等学校—教材 Ⅳ. TP36

中国版本图书馆CIP数据核字(2003)第052344号

书　　名：微机原理与接口技术教程
著作责任者：王克义 鲁守智 蔡建新 王文保 编著
责 任 编 辑：沈承凤
标 准 书 号：ISBN 7-301-06367-9/TP・0714
出 版 发 行：北京大学出版社
地　　址：北京市海淀区中关村 北京大学校内 100871
网　　址：http://cbs.pku.edu.cn 电子信箱：zpup@pup.pku.edu.cn
电　　话：邮购部62752015 发行部62750672 编辑部62752038
排 版 者：兴盛达打字服务社 82715400
印 刷 者：涿州市星河印刷有限公司
经 销 者：新华书店
787毫米×1092毫米 16开本 49印张 1217千字
2004年12月第1版 2004年12月第1次印刷
定　　价：57.00元

前　言

“微机原理”是高等学校理工科学生的一门重要的计算机基础课程，也是理工科大学生学习和掌握计算机科学技术基础、汇编语言程序设计及常用接口技术的入门课程。通过本课程的学习，可以使学生从理论和实践上掌握微型计算机的基本组成和工作原理，建立微机整机概念，具备利用微机技术进行硬、软件开发的初步能力。学习本课程对于熟悉和掌握现代计算机的基本概念和技术以及学习后续有关计算机课程（如计算机体系结构、操作系统、计算机网络、嵌入式系统等）均具有重要意义。本书是该课程使用的基本教材。

本书以80x86/Pentium系列微型计算机为背景机，全面、系统地介绍了微型计算机的基本结构、工作原理及典型接口技术。全书共21章，按内容可分为六个知识单元：①计算机科学技术基础（第1、2、3、4、5章）；②指令系统及汇编语言程序设计（第7、8、9章）；③微处理器结构及工作原理（第6、10）章；④存储器及其接口（第12章）；⑤I/O接口及外设（第11、13、14、15、16、17、18、21章）；⑥高性能微处理器及相关技术（第19、20章）。

与其他同类教材不同的是，本教材在第1～5章的计算机科学技术基础部分有意识地融入了传统的“计算机组成原理”课程的有关重点内容，如计算机运算器的基本结构及运算方法、控制器的基本结构及控制方法、硬布线控制器及微程序控制器的实现技术、定点数及浮点数的表示及四则运算方法等，并在第6章简要介绍微处理器的编程结构之后，用第7、8、9三章的篇幅翔实地讲解了80x86/Pentium的指令系统及汇编语言程序设计的基本概念与方法；后续章节则相继介绍了微处理器的内部与外部结构、存储器及其接口、输入/输出及可编程接口电路以及高性能微处理器的相关概念和技术；其中，在第19章中，专门介绍了80x86/Pentium保护模式的有关概念和结构，如描述符及描述符表机制、保护模式的存储管理和地址转换以及多任务的实现及保护机制等；在第20章中，重点介绍了现代高性能微处理器的多项先进技术和典型结构，如指令级并行及流水线中的“相关”概念及其处理技术、超标量流水线技术、超长指令字结构、RISC技术、MMX技术及多处理器并行结构等；最后还简要介绍了微机外部设备的有关知识和技术。

本教材可供54～72学时的课堂教学使用。教师在使用本教材时，可根据教学大纲的具体要求，进行灵活安排。如对于已学过数字逻辑电路的学生，可略去第4章不讲而直接进入其他有关章节的讲解；另外，对于有些章节的内容（如指令系统）可不完全通过课堂讲授而是通过上机实习及接口实验等教学环节的带动来完成。在本书中，打*号的部分是作者认为可取、舍的内容。

此外，鉴于本课程是技术性、实践性较强的课程，所以应有相应的上机及实验环节。教师可根据具体的实验设备及上机条件，安排适当的汇编上机及接口实验内容。

书后给出了大部分习题的参考答案或解题思路，供读者练习和自我检测之用。

本书由北京大学王克义、鲁守智、蔡建新、王文保四位编著者合作完成。王克义负责全书统稿工作，并编写第6、10、11、12、13、14、16、18、19、20章，鲁守智编写第1、2、3、4、5、7、8、9章，蔡建新编写第15、17章，王文保编写第21章。赵引和张延娜两位老师参加了本书相关章节初稿的编写及审校工作，提出了许多有益的修改意见。

本书的作者都是多年来在教学第一线主讲本课程的教师。本书是在作者近年来给北京大学计算机系本科生以及北京大学理科非计算机专业本科生教学实践的基础上编写而成的。在编写过程中，参考和吸收了国外较新同类教材及国内兄弟院校优秀教材的有关内容，在此向有关作者一并致谢。

在本书的编写、出版过程中，得到北京大学信息科学技术学院领导的大力支持。北大出版社沈承凤高级工程师为本书的出版付出了艰辛和智慧。在此，谨向他们表示深深的感谢。

由于编著者水平所限，书中一定存在不少差错和疏漏，诚请广大读者及专家批评指正。

编著者

2004年10月于北京大学

目　录

第 1 章　进位计数制及数的转换

1.1　进位计数制及其基数和权

进位计数制(简称进位制)是一种表示数及计数的方法,是指从有限几个基本数字符号中取数字符号,用按位置定值的方法将取得的数字符号排列起来表示数,用计满后向高位进位的方法计数。这样表示数及计数时,每个数字符号所代表的值既取决于数字符号本身也取决于它所在的位置,这样做的好处是只用有限几个基本数字符号就能表示出无限多个数。用于计数的数字符号称为数码。

例如,在十进位制中,若表示二百零五点三七五这个数,只需从0、1、2、3、4、5、6、7、8、9十个基本数码中取出2、0、5、3、7、5,并从左到右将每个数码分别按乘以百、十、个、十分之一、百分之一、千分之一的顺序排列,表示成205.375即可。在计数时,如果某一位计到十就要变成0并要在其左边一位上加1,这些我们都是很熟悉的。

在计算机技术中,常用的进位制除了十进位制(简称十进制)以外还有二进位制(简称二进制)、十六进位制(简称十六进制)和八进位制(简称八进制)。

任何一种进位制都能用有限几个基本数码表示出所有的无符号整数,我们称这几个基本数码为进位制的基。基是指在进位制中允许使用的数码。基的个数称为基数。不同进位制的基和基数都不同。

任何一种进位制都是用基本数码的排列表示数,同一数码位于数的不同数位时,代表的数值不同,我们称位于不同数位上的数码有不同的位权,简称权。权是指进位制中基数的各整数次幂。幂指数是数码位置的序号:小数点左边第一个数码序号为0,向左各位的序号依次为1、2、3、…,向右各位的序号依次为-1、-2、-3、…。每一位数码都有惟一确定的权,不同位置的数码有不同的权,权值较大的数码称为高位数码,权值较小的数码称为低位数码,每个数码所代表的数值等于该数码与其权值的乘积。

在任何一种进位制中,任意一个无正、负号的数都可写成

$$a_{n-1}a_{n-2}\cdots a_1a_0\ .\ a_{-1}a_{-2}\cdots a_{-(m-1)}a_{-m} \tag{1.1}$$

的形式,其中a_{n-1}、a_{n-2}、…、a_1、a_0、a_{-1}、a_{-2}、…、$a_{-(m-1)}$、a_{-m}均为数码,它们都是该进位制中允许使用的基本数码中的某一个。下标n、m为无正、负号的整数,n为(1.1)式中整数部分的位数,m为(1.1)式中小数部分的位数。

在后面的讨论中,我们将无正、负号的数称为无符号数,将带正、负号的数称为带符号数。无符号数和带符号数应该严格区别,它们在计算机内的表示形式是不同的。

1.1.1　十进制

在十进制中,(1.1)式的含义是

$$a_{n-1}\times 10^{n-1}+a_{n-2}\times 10^{n-2}+\cdots+a_0\times 10^0+a_{-1}\times 10^{-1}+a_{-2}\times 10^{-2}+\cdots+a_{-m}\times 10^{-m} \tag{1.2}$$

其中，每一个$a_i(i=n-1、n-2、\cdots、1、0、-1、-2、\cdots、-m)$都是0、1、2、3、4、5、6、7、8、9十个数码中的某一个。例如，十进制数205.375的含义用十进制数表示就是

$$2\times10^2+0\times10^1+5\times10^0+3\times10^{-1}+7\times10^{-2}+5\times10^{-3}$$

在十进制中，使用10个数码就能表示出所有的无符号整数，因此，十进制的基为0、1、2、3、4、5、6、7、8、9共10个数码，基数为10，计数时逢10进位。

十进制数中各数码的权为10的整数次幂，按式(1.1)的表示法，数码a_i的权是$10^i(i=n-1、n-2、\cdots、1、0、-1、-2、\cdots、-m)$，式(1.2)就是式(1.1)按十进制权的展开式。

常见的十进制权与十进制数的对应关系如表1-1所示。

表1-1　常见的十进制权与十进制数的对应关系

10^6	10^5	10^4	10^3	10^2	10^1	10^0
1000000	100000	10000	1000	100	10	1
10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
.1	.01	.001	.0001	.00001	.000001	.0000001

用n位十进制数码能组成10^n个不同的数。例如，用4位十进制数码能组成

0000　0001　0002　…　9998　9999

共$10^4=10000$个不同的数。

在十进制中，

$$1K（开，一千）=10^3=1000$$
$$1M（兆，百万）=10^6=1000K=1000000$$
$$1G（吉，十亿）=10^9=1000M=1000000000$$
$$1T（太，万亿）=10^{12}=1000G=1000000000000$$

通常用字母D或d(Decimal)标识十进制数，有时也用下标$_{10}$或下标$_{十}$标识十进制数，对十进制数也可以不做任何标记。例如，

$$205.375_{10}=205.375_{十}=205.375D=205.375d=205.375$$

1.1.2　二进制

在二进制中，(1.1)式的含义是

$$a_{n-1}\times2^{n-1}+a_{n-2}\times2^{n-2}+\cdots+a_0\times2^0+a_{-1}\times2^{-1}+a_{-2}\times2^{-2}+\cdots+a_{-m}\times2^{-m}\qquad(1.3)$$

其中，每一个$a_i(i=n-1、n-2、\cdots、1、0、-1、-2、\cdots、-m)$都是0、1两个数码中的某一个，例如，二进制数11001101.011的含义用十进制数表示就是

$$1\times2^7+1\times2^6+0\times2^5+0\times2^4+1\times2^3+1\times2^2+0\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2}+1\times2^{-3}$$

在二进制中，使用两个数码就能表示出所有的无符号整数，因此，二进制的基为0、1两个数码，基数为2，计数时逢2进位。

二进制数中各数码的权为2的整数次幂，按式(1.1)的表示法，数码a_i的权是$2^i(i=n-1、n-2、\cdots、1、0、-1、-2、\cdots、-m)$，式(1.3)就是式(1.1)按二进制权的展开式。

常见的二进制权与十进制数的对应关系如表1-2所示。

表 1-2　常见的二进制权与十进制数的对应关系

2^{20}	2^{19}	2^{18}	2^{17}	2^{16}	2^{15}	2^{14}
1048576	524288	262144	131072	65536	32768	16384
2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}
8192	4096	2048	1024	512	256	128
2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
64	32	16	8	4	2	1
2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
.5	.25	.125	.0625	.03125	.015625	.0078125

用n位二进制数码能组成2^n个不同的数。例如，用4位二进制数码能组成

0000　0001　0010　0011　0100　0101　0110　0111

1000　1001　1010　1011　1100　1101　1110　1111

共$2^4=16$个不同的数，与它们等值的十进制数依次是

0　1　2　3　4　5　6　7

8　9　10　11　12　13　14　15

在二进制中，习惯上认为

1K (开，一千)$=2^{10}=1024$

1M (兆，百万)$=2^{20}=1024K=1048576$

1G (吉，十亿)$=2^{30}=1024M=1073741824$

1T (太，万亿)$=2^{40}=1024G=1099511627776$

通常用字母B或b(Binary)标识二进制数，有时也用下标$_2$或下标$_{二}$标识二进制数。例如，

$$11001101.011_2 = 11001101.011_{二} = 11001101.011B = 11001101.011b$$

1.1.3　十六进制

在十六进制中，(1.1)式的含义是

$$a_{n-1}\times16^{n-1}+a_{n-2}\times16^{n-2}+\cdots+a_0\times16^0+a_{-1}\times16^{-1}+a_{-2}\times16^{-2}+\cdots+a_{-m}\times16^{-m} \qquad (1.4)$$

其中，每一个a_i($i=n-1$、$n-2$、…、1、0、−1、−2、…、$-m$)都是0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F共16个数码中的某一个，这里，用A、B、C、D、E、F(或者a、b、c、d、e、f)分别代表十进制数10、11、12、13、14、15。例如，十六进制数CD.6的含义用十进制数表示就是

$$12\times16^1+13\times16^0+6\times16^{-1}$$

在十六进制中，使用16个数码就能表示出所有的无符号整数，所以，十六进制的基为0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F共16个数码，基数为16，计数时逢16进位。

十六进制数中各数码的权为16的整数次幂，按式(1.1)的表示法，数码a_i的权是16^i($i=n-1$、$n-2$、…、1、0、−1、−2、…、$-m$)，式(1.4)就是式(1.1)按十六进制权的展开式。

表 1-3　常见的十六进制权与十进制数的对应关系

16^5	16^4	16^3	16^2	16^1	16^0	16^{-1}	16^{-2}
1048576	65536	4096	256	16	1	.0625	.00390625

常见的十六进制权与十进制数的对应关系如表1-3所示。

用n位十六进制数码能组成16^n个不同的数。例如,用4位十六进制数码能组成

0000　0001　0002　…　0009　000A　000B　…　000E　000F

0010　0011　0012　…　0019　001A　001B　…　001E　001F

…………………………

FFF0　FFF1　FFF2　…　FFF9　FFFA　FFFB　…　FFFE　FFFF

共$16^4=65536$个不同的数,与它们等值的十进制数依次是

0　1　2　…　9　10　11　…　14　15

16　17　18　…　25　26　27　…　30　31

…………………………

65520　65521　65522　…　65529　65530　65531　…　65534　65535

通常用字母H或h(Hexadecimal)标识十六进制数,有时也用下标$_{16}$或下标$_{十六}$标识十六进制数。例如,

$$CD.6_{16}=CD.6_{十六}=CD.6H=CD.6h$$

对以A、B、C、D、E、F之一开头的十六进制数,必要时应在其左边加一个0,以便区别于由字母开头的用字母数字串组成的名字。例如,数CD.6H必要时应写为0CD.6H。

十六进制数码与二进制数码之间有简单的对应关系,参见表1-4。

表1-4　十六进制数码与二进制数码之间的对应关系

0	1	2	3	4	5	6	7
0000	0001	0010	0011	0100	0101	0110	0111
8	9	A	B	C	D	E	F
1000	1001	1010	1011	1100	1101	1110	1111

表1-4指出,4位二进制数码的所有不同组合与全部十六进制数码之间是一一对应的,1位十六进制数码相当于4位二进制数码。

1.1.4　R(R为整数且R>1)进制、八进制

在R进制中,(1.1)式的含义是

$$a_{n-1}\times R^{n-1}+a_{n-2}\times R^{n-2}+\cdots+a_0\times R^0+a_{-1}\times R^{-1}+a_{-2}\times R^{-2}+\cdots+a_{-m}\times R^{-m} \quad (1.5)$$

其中,每一个$a_i(i=n-1、n-2、\cdots、1、0、-1、-2、\cdots、-m)$都是0、1、… 共R个不同数码中的某一个。

在R进制中,使用R个数码就能表示出所有的无符号整数,所以,R进制的基数为R,计数时逢R进位。

R进制数中各数码的权为R的整数次幂,按式(1.1)的表示法,数码a_i的权是$R^i(i=n-1、n-2、\cdots、1、0、-1、-2、\cdots、-m)$,式(1.5)就是式(1.1)按R进制权的展开式。

用n位R进制数码能组成R^n个不同的数。

在R进制中,令R=10就得到十进制,令R=2就得到二进制,令R=16就得到十六进制。换句话说,十进制、二进制、十六进制都是R进制的特例。

在R进制中,令R=8就得到八进制。八进制的基为0、1、2、3、4、5、6、7共8个数码,基数为

8，计数时逢8进位。八进制数中各数码的权为8的整数次幂。用n位八进制数码能组成8^n个不同的数。通常用字母O或o(Octal)、字母Q或q标识八进制数，有时也用下标$_8$或下标$_八$标识八进制数。例如，

$$315.3_8 = 315.3_{八} = 315.3\text{O} = 315.3\text{o} = 315.3\text{Q} = 315.3\text{q}$$

很明显，3位二进制数码的所有不同组合与全部八进制数码之间是一一对应的，1位八进制数码相当于3位二进制数码。

1.2 R(R为整数且R>1)进制数的四则运算

R进制数的四则运算方法与十进制数相同，只须注意逢R进1，借1当R即可。

例1.1 二进制数的四则运算，以整数为例。

(1) 1101+1011=11000

```
   1101
+  1011
-------
  11000
```

(2) 11000−1011=1101

```
  11000
-  1011
-------
   1101
```

(3) 1101×1011=10001111

```
       1101
×      1011
-----------
       1101
      1101
     0000
+   1101
-----------
   10001111
```

(4) 10001111÷1101=1011

```
          1011
      ---------
1101 ) 10001111
       1101
       ----
        1001
        0000
        -----
        10011
         1101
         -----
          1101
          1101
          ----
             0
```

显然，二进制数的运算比其他任何进制数的运算都简单：只有0、1两个数码间的4种(0和0,0和1,1和0,1和1)基本运算。做乘法时，乘以1就是以乘数的位置为最低位将被乘数重写一遍，4位数乘以4位数最多得8位数，最多需要做8位数相加的操作。做除法时，够除时商写1，不够除时商写0，除不尽时将得到无限循环小数。

在乘法的例子中，参与求和的4个加数除了零以外都是被乘数左移后的值，所以，乘法实际上是连续的加法，可通过相加和移位实现。在除法的例子中，每次从被除数或余数中减去的数除了零以外都是除数右移后的值，所以，除法实际上是连续的减法，可通过相减和移位实现。由此可见，在二进制中，加、减、乘、除四则运算都可通过相加、相减和移位三种操作来完成。

可以验证，乘以2等于将被乘数相对于小数点左移1位，低位补0；除以2等于将被除数相对于小数点右移1位，高位补0。

例1.2 十六进制数的四则运算，以整数为例。

(1) C7A5H+F5E3H=1BD88H

```
   C7A5
+  F5E3
-------
  1BD88
```

(2) 1BD88H−F5E3H=C7A5H

```
  1BD88
-  F5E3
-------
   C7A5
```

*(3) 7ABH×6FH=35325H

```
     7AB
×     6F
--------
    7305
+  2E02
--------
   35325
```

*(4) 35325H÷7ABH=6FH

```
          6F
     ---------
7AB | 35325
      2E02
      ----
       7305
       7305
       ----
          0
```

可以验证，乘以16 等于将十六进制的被乘数相对于小数点左移1 位，低位补1 个0，相当于将二进制的被乘数相对于小数点左移4 位，低位补4 个0；除以16 等于将十六进制的被除数相对于小数点右移1 位，高位补1 个0，相当于将二进制的被除数相对于小数点右移4 位，高位补4 个0。

*1.3 二进制数的按位四则运算

按位二进制四则运算就是没有进位或借位的二进制四则运算，运算时将参与运算的二进制数的每一位都作为一个独立的数码看待，位和位之间互相没有联系。

因为一个n 位二进制数

$$a_{n-1}a_{n-2}\cdots a_1 a_0$$

和一个2 的n－1 次多项式（即按二进制权的展开式）

$$a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\cdots+a_1x+a_0 \quad (x=2)$$

之间是一一对应的，所以，二进制数的按位四则运算可通过对应多项式的按位四则运算实现，多项式的按位四则运算也可通过对应二进制数的按位四则运算实现。

1.3.1 按位加法

此法就是按位相加，不进位，即0＋0＝0，0＋1＝1，1＋0＝1，1＋1＝0，也就是“相同得0，不同得1”，或者说，“加0 时不变，被0 加时不变；加1 时变反，被1 加时变反”。例如，

1100＋1010＝0110

```
   1100
+  1010
-------
   0110
```

$(x^3+x^2)+(x^3+x)=(x^2+x)$

$$\begin{array}{r} x^3+x^2 \\ +\,x^3 \qquad +x \\ \hline x^2+x \end{array}$$

为了与普通加法相区别，按位加法可用符号“⊕”表示，例如，1100＋1011 表示普通加法，1100⊕1011 表示按位加法。

多个二进制位依次按位加时，若有偶数个1 则结果为0，若有奇数个1 则结果为1。例如，

$$0\oplus 0\oplus 1\oplus 0\oplus 1=0 \qquad 1\oplus 0\oplus 1\oplus 1\oplus 0=1$$

1.3.2 按位减法

此法就是按位相减，不借位，即0－0＝0，0－1＝1，1－0＝1，1－1＝0，也就是“相同得0，不同得1”，或者说，“减0 时不变，被0 减时不变；减1 时变反，被1 减时变反”。

为了与普通减法相区别，按位减法可用符号“⊖”表示，例如，1100－1011 表示普通减法，1100⊖1011 表示按位减法。

不难看出，按位减法和按位加法的结果是相同的，因此，按位减可用按位加代替，按位加也

可用按位减代替。还可以看出，按位减法与按位加法互为逆运算。

1.3.3 按位乘法

此法就是用按位加求部分积之和，不进位。例如，

$1101\times1011=1111111$　　　$(x^3+x^2+1)\times(x^3+x+1)=x^6+x^5+x^4+x^3+x^2+x+1$

$$
\begin{array}{lr}
 & 1101 \\
\times & 1011 \\ \hline
 & 1101 \\
 & 1101 \\
 & 0000 \\
\oplus & 1101 \\ \hline
 & 1111111
\end{array}
\qquad
\begin{array}{lr}
 & x^3+x^2+1 \\
\times & x^3+x+1 \\ \hline
 & x^3+x^2+1 \\
 & x^4+x^3+x \\
\oplus & x^6+x^5+x^3 \\ \hline
 & x^6+x^5+x^4+x^3+x^2+x+1
\end{array}
$$

我们看到，按位乘法中相乘的运算和普通的乘法相同，只是其中的加法运算采用按位的算法。

1.3.4 按位除法

此法就是用按位减求部分余数，不借位。相除时上商的规则是：余数首位为1时商上1，余数首位为0时商上0。每得一位商应使部分余数减少一位，当余数位数小于除数位数时即得最后的余数。例如，

10000÷101=101，余 01　　　$(x^4)\div(x^2+1)=(x^2+1)$，余 1

$$
\begin{array}{rr}
 & 101 \\ \cline{2-2}
101) & 10000 \\
\ominus & 101 \\ \hline
 & 010 \\
\ominus & 000 \\ \hline
 & 100 \\
\ominus & 101 \\ \hline
 & 01
\end{array}
\qquad
\begin{array}{rr}
 & x^2+1 \\ \cline{2-2}
x^2+1) & x^4 \\
\ominus & x^4\ +x^2 \\ \hline
 & x^2 \\
\ominus & \\ \hline
 & x^2 \\
\ominus & x^2+1 \\ \hline
 & 1
\end{array}
$$

我们看到，按位除法中相除的运算和普通的除法相同，只是其中的减法运算采用按位的算法。

可以验证，按位除法与按位乘法互为逆运算。

1.4 不同基数的数之间的转换

同一个数值在不同基数下的表示形式不同，这些不同的形式之间一定可互相转换。我们将被转换的数称为源进位制中的数，简称源数，转换后的数称为目的进位制中的数，简称目的数。

1.4.1 R(R为整数且R＞1)进制数转换为十进制数

1. *方法1：乘以权值求和法*

此法也称按权展开求和法。就是将(源数的)各位数码乘以各自的权值，写成十进制数，然

后再求和。也就是算出(源数的)每一位数码所代表的十进制数，再将这些十进制数相加。

例1.3 11001101.011B $=10000000B+1000000B+1000B+100B+1+0.01B+0.001B$

$=1\times2^7+1\times2^6+1\times2^3+1\times2^2+1\times2^0+1\times2^{-2}+1\times2^{-3}$

$=128+64+8+4+1+0.25+0.125$

$=205.375$

例1.4 0CD.6H $=0C0H+0DH+0.6H$

$=12\times16^1+13\times16^0+6\times16^{-1}$

$=192+13+0.375$

$=205.375$

例1.5 $(102.2)_3=(100)_3+(2)_3+(0.2)_3$

$=1\times3^2+2\times3^0+2\times3^{-1}$

$=9+2+0.666\cdots$

$=11.666\cdots$

*2. 方法2：乘(除)以基数累加法

整数部分：从高位数码起，不断地乘以(源数的)基数再加上低一位数码，直至加上最低整数位数码止。

小数部分：从低位数码起，不断地除以(源数的)基数再加上高一位数码，直至加上最高小数位数码后除以(源数的)基数止。

例1.6 11001101.011B=205.375，算式如下：

整数部分： 11001101

$=((((((1\times2+1)\times2+0)\times2+0)\times2+1)\times2+1)\times2+0)\times2+1$

$=(((((3\times2+0)\times2+0)\times2+1)\times2+1)\times2+0)\times2+1$

$=((((6\times2+0)\times2+1)\times2+1)\times2+0)\times2+1$

$=(((12\times2+1)\times2+1)\times2+0)\times2+1$

$=((25\times2+1)\times2+0)\times2+1$

$=(51\times2+0)\times2+1$

$=102\times2+1$

$=205$

小数部分： 0.011

$=((1\times2^{-1}+1)\times2^{-1}+0)\times2^{-1}$

$=(1.5\times2^{-1}+0)\times2^{-1}$

$=0.75\times2^{-1}$

$=0.375$

例1.7 0CD.6H=205.375，算式如下：

整数部分： 0CDH

$=12\times16+13$

$=192+13$

$=205$

小数部分： 0.6H

$=6\times16^{-1}$

$=6\times0.0625$

$=0.375$

例1.8 $(102.2)_3=11.666\cdots$，算式如下：

整数部分： $(102)_3$

$=(1\times3+0)\times3+2$

$=11$

小数部分： $(0.2)_3$

$=2\times3^{-1}$

$=0.666\cdots$