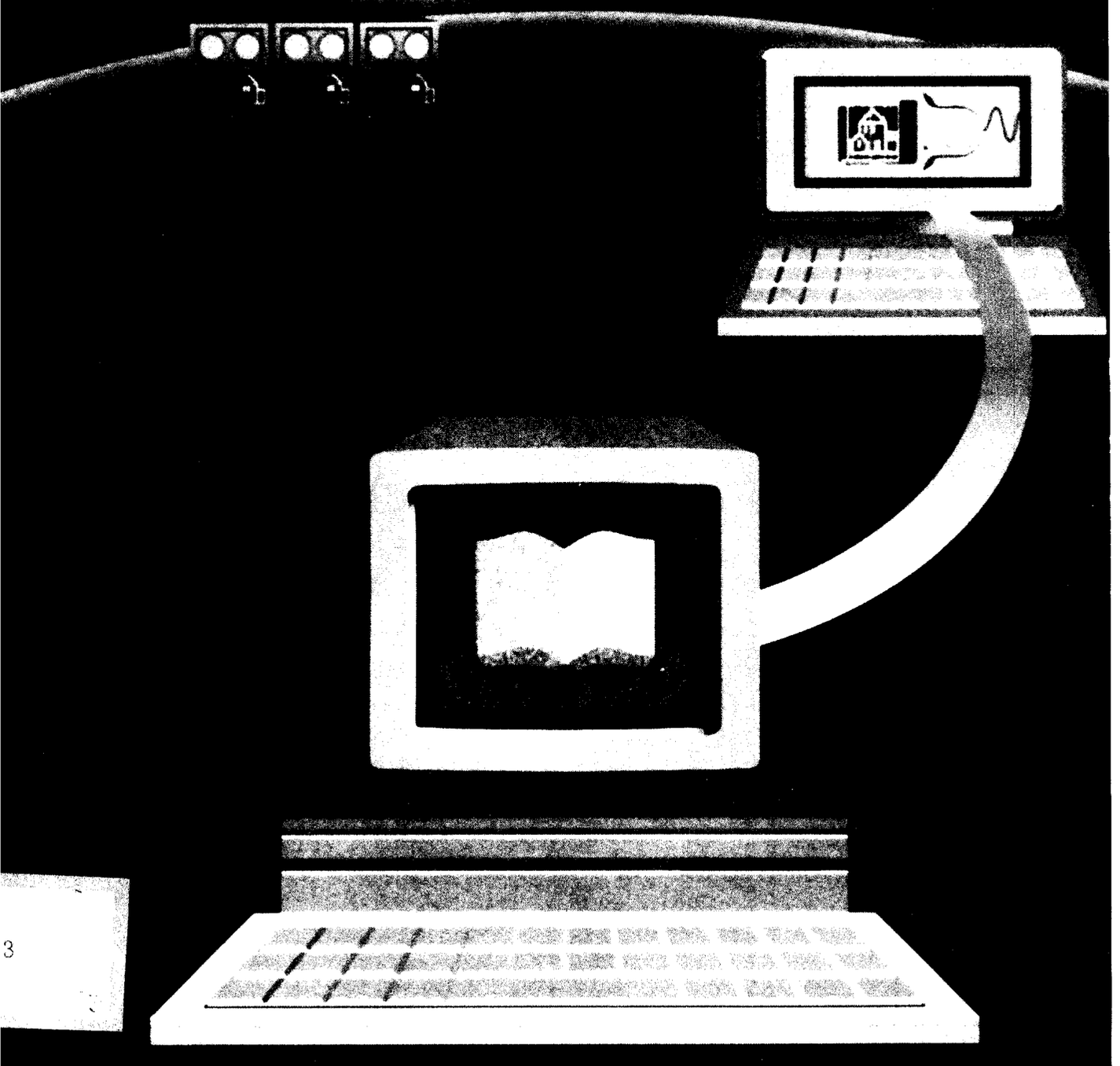
汇编语言基础及CASL汇编语言

刘克武 刘 轩 郭怀峥



北京大学出版社

内 容 简 介

本书分为两大部分。第一部分系统地介绍了使用汇编语言所涉及到的计算机软、硬件基础。内容包括计算机与数制；数在计算机中的表示；数在计算机中的编码；信息在计算机中的代码；计算机中的逻辑运算与操作等。第二部分全面地介绍了目前在国际上采用的，建立在假想机上的CASL汇编语言；而CASL汇编语言已列为我国软件人员水平考试程序员级可选语言之一，也是高级程序员级的必考语言。

通过本书学习既可以打下良好的计算机基础，又可以掌握一门基础的汇编语言，为学习其它汇编语言奠定坚实的基础。

本书可作为大专院校计算机专业教材，也可以作为软件人员水平考试培训教材。

图书在版编目（CIP）数据

汇编语言基础及CASL汇编语言/刘克武编著．—北京：北京大学出版社，1997
ISBN 7-301-03591-8

Ⅰ．汇… Ⅱ．刘… Ⅲ．①汇编程序-程序语言-基本知识 ②汇编程序-CASL程序语言 Ⅳ．TP312

中国版本图书馆CIP数据核字（97）第26750号

书　　名：汇编语言基础及CASL汇编语言
著作责任者：刘克武等
责 任 编 辑：郭佑民
标 准 书 号：ISBN 7-301-03591-8/TP・379
出　版　者：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
电　　话：出版部 62752015　发行部 62559712　编辑部 62752032
排　印　者：国防科工委印刷厂印刷
发　行　者：北京大学出版社
经　销　者：新华书店
787×1092　16开本　12.75印张　320千字
1998年2月第一版　1998年2月第一次印刷
定　　价：21.00元

前　言

汇编语言是深入到计算底层，进行软件开发及应用的必不可少的语言工具。在计算机的发展与应用过程中，汇编语言在系统软件设计、数据采集、自动控制、图形、图像处理及联机实时处理中一直在发挥着它的作用。

汇编语言属于低级语言，它对计算机的硬件环境有很大的依赖性。因此，使用汇编语言因机器不同而异，机器的更新，汇编语言也随之更新。这种状况，一方面给学习、使用汇编语言者带来了诸多的困难；另一方面也使得针对一种机型介绍汇编语言的书籍时效很短。如何使学习汇编语言有长时间的继承性，如何使汇编语言的书籍也像高级语言一样，具有较长的时效性，这是编写本书的初衷。

本书分为两部分，共十三章。第一章到第五章是以学习汇编语言为纲，介绍计算机的软、硬件基础。内容包括：计算机与二进制；数的存、取空间及表示形式；数在计算机中的编码；计算机与信息代码；计算机中的逻辑运算与操作。第六章到第十三章系统地介绍了建立在假想机上的，名为CASL的汇编语言。CASL汇编语言集中了汇编语言中的主要基础，在美国、日本等国家广泛用于汇编语言的教学中。在我国也把CASL汇编语言作为“计算机软件专业技术资格和水平考试”所使用的汇编语言。CASL汇编语言是高级程序员级必考科目，也是程序员级任选五种考试语言之一。

本书选材精炼、系统；叙述由浅入深、前后连贯，参阅了不少有关书籍并结合多年教学经验编写而成。刘畅、郭怀峥参加了本书第六章到第十三章的部分编写工作。

本书可作为大专院校计算机及相关专业教授汇编语言的教材，也可以作为软件人员水平考试培训教材。

书中不足之处，恳请专家和读者予以指正。

编　者

1997年7月

目　录

第一章　计算机与二进制

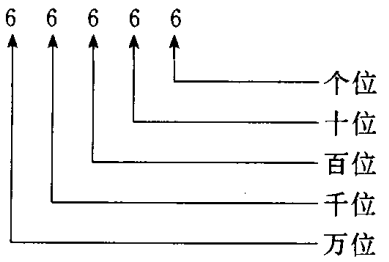
§1.1　十进制与二进制

一、十进制

在计数和计量中有很多种进制，人们使用最多的一种计数制是十进制。

1. 十进制的权及权系数

十进制是一种有0—9十个基数字，逢十进一的计数制。基数字0—9的任意排列就构成了任意位的十进制数，例如，389，476，584等都是十进制数。任意一个十进制整数其右端的一位定义为最低位，其左端的一位定义为最高位，于是任意一个十进制数，从右至左就出现了个位、十位、百位、千位、万位的称呼。例如，

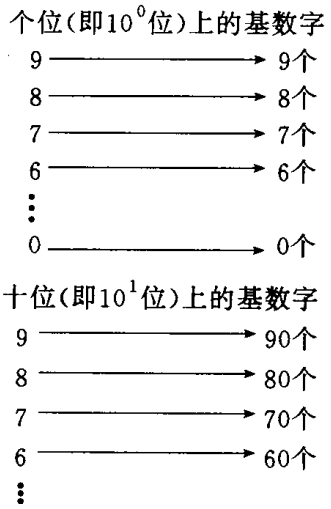


分析此例不难发现，同一个基数字6在它构成十进制数时，由于它所处的位置不同，而它所代表的值也不一样，这个原则就是十进制的位权，简称权。

十进制的权可以用10的指数形式来表示，例如：

十进制的权（n,m均为正整数）									
……	万位	千位	百位	十位	个位	十分位	百分位	千分位	……
10^{n}…	10^{4}	10^{3}	10^{2}	10^{1}	10^{0}	10^{-1}	10^{-2}	10^{-3}	…10^{-m}
← 整数部分 →					小数点	← 小数部分 →			

在十进制中还可以发现另外一个原则，即在同一个位权上，基数字不同其值也不一样。例如：



由此说明，十进制的每一位上都可以出现代表不同值的0—9，而把0—9称为权系数 。

（2）十进制数的多项式表示

十进制的位权和权系数决定着一个十进制数的大小，因此任意一个十进制数都可以用位权及权系数表示出来。例如，

$$\begin{aligned}53429=&50000+\\&3000+\\&400+\\&20+\\&9\\\hline&53429\end{aligned}$$

用位权和权系数改写该数为：

$53429=5\times10^4+3\times10^3+4\times10^2+2\times10^1+9\times10^0$

由此可以说明，对于一个十进制数，可以写成位权与权系数乘积之和所表示出的多项式。十进制数的这一性质是具有普遍意义的，可以推广到任意一个十进制数上。

设 S 为任意的一个十进制数，S 的每一位用 x_i 表示。显然，x_i 为正整数，且满足 $0\leqslant x_i\leqslant 9$；其中 $i=n$，…，9，8，7，6，5，4，3，2，1，0，-1，…，$-m$。

这样，任意一个十进制数都可以表示成权系数 x 的序列。即，

$$S=x_n\cdots x_1x_0\cdot x_{-1}x_{-2}\cdots x_{-m}$$

将其写成位权与权系数乘积之和的多项式为，

$$S=x_n\times10^n+\cdots+x_1\times10^1+x_0\times10^0+x_{-1}\times10^{-1}+x_{-2}\times10^{-2}+\cdots+x_{-m}\times10^{-m}$$

也可以用求和形式表示为：

$$S=\sum_{i=-m}^{n}x_i\cdot10^i$$

其中，x_i 为权系数，10^i 为位权。

从以上分析可知，任何一个十进制数都可以写成权系数与位权乘积之和的多项式。由于这个多项式表示的是逢十进一的十进制数，所以它是以10为底幂的多项式。多项式的每一项都有位权，因此交换多项式各项的位置不会影响十进制数的值。该多项式可以用升幂排列，也可以用降幂排列。十进制数的多项式表示可以推广到其它进制，而数的多项式表示是各种进制相互转换的基础。

二、二进制

在计算机中采用的不是十进制而是二进制，其原因我们将在后面分析。首先，我们仿照分析十进制的方法来分析二进制。

1. 二进制及二进制数

二进制是一种只有0和1两个基数字，逢二进一的计数制。二进制的基数字0，1的任意排列就构成了二进制数，借助小数点、正、负号可以表示出二进制的整数、小数、正数、负数。例如，

（1）二进制正整数：1011_2，1100_2，1111_2，100001_2。

（2）二进制负整数：-101_2，-110_2，-1101_2，-100_2。

（3）含有小数的二进制数：11.11_2，-10.01_2，0.111_2。

(4) 含有指数的二进制数：$(111)_2^{10}$，$(-11)_2^{11}$，$(0.1)_2^{11}$

为了区分二进制数与十进制数，可以在数的右下角加注下角标。例如，100_2 表示二进制；100_{10}表示十进制数。

2. 二进制的权及权系数

二进制也是一种有权进位制。权的含义是指同一个基数字，在用其组成二进制数时，由于它所处的位置不同，其值也不一样。二进制的权可以用以 2 为底的指数形式来表示。见下表。

二进制的权（n、m均为正整数）							
$2^n\cdots$	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	$\cdots2^{-m}$
← 整数部分 →				小数点	← 小数部分 →		

二进制的权系数只有 0 和 1 两个，当某位的权系数为 0 时，则本位的值为 0；当某位的权系数为 1 时，其值为 2 的某次方。

3. 二进制数的多项式表示

任何一个二进制数，也和十进制数一样可以表示成多项式的形式。例如，

$$\begin{aligned}11111_2=&10000+\\&1000+\\&100+\\&10+\\&1\\\hline &11111_2\end{aligned}$$

二进制数的形式可以写成权系数与位权乘积之和，下面看几个例子。

【例 1】 将二进制数 1101_2 写成多项式形式：

$1101_2=1\times2^3+1\times2^2+0\times2^1+1\times2^0$。

【例 2】 将二进制数 0.1101_2 写成多项式形式：

$0.1101_2=1\times2^{-1}+1\times2^{-2}+0\times2^{-3}+1\times2^{-4}$。

【例 3】 将二进制数 111.111_2 写成多项式形式：

$111.111_2=1\times2^2+1\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}+1\times2^{-3}$。

分析以上三例可以看出，由于二进制数的权系数不为 1 则为 0，所以当权系数为 1 时，1 乘位权仍等于位权；当权系数为 0 时，0 乘位权其值仍为 0。因此，二进制数的多项式可以表示成位权之和的形式。见下例。

【例 4】 用位权之和写出二进制数 1111_2：

$1111_2=2^3+2^2+2^1+2^0$。

【例 5】 用位权之和写出二进制数 100.001_2：

$100.001_2=2^2+2^{-3}$。

由例 5 可以看出，一个二进制数可以写成权系数为 1 所对应的位权之和所表示出的多项式。见下例。

【例 6】 将 0.00011_2 表示成多项式：

$0.00011_2=2^{-4}+2^{-5}$。

【例 7】 将 1010.101_2 表示成多项式：

$1010.101_2=2^3+2^1+2^{-1}+2^{-3}$。

对于任意位的二进制数，用 R 表示。R 的每一位用 x_i 表示，则二进制数 R 可以表示为：

$$R=x_n\times 2^n+\cdots+x_2\times 2^2+x_1\times 2^1+x_0\times 2^0+x_{-1}\times 2^{-1}+x_{-2}\times 2^{-2}+\cdots+x_{-m}\times 2^{-m}$$

写成求和的形式为：

$$R=\sum_{i=-m}^{n} x_i\cdot 2^i$$

其中，x_i 为 1 或为 0，$i=n$，…，9，8，7，6，5，4，3，2，1，0，-1，…，$-m$。

三、二进制数转换成十进制数

1. 转换原理

度量同一事物，不管是采用二进制还是十进制，其结果应该是相等的。这个道理是很容易理解的。例如教室里的电灯，我们用十进制数电灯的个数时为 6；用二进制数个数时为 110。如果数的个数都是对的，那么必然有 $6_{10}=110_2$。我们用 S 表示十进制数，用 R 表示二进制数，显然，

$$S=R$$

对于该式可以进一步推广，即任何一个二进制数 R，必然会找到一个与之相等的十进制数 S。这就是二进制数转换为十进制数的基础。

2. 转换法则

我们从 $S=R$ 出发，假定 $R=110$（即电灯个数），则 $S=110_2$，把二进制数 110 写为多项式形式，则有：

$$S=R=110_2=1\times 2^2+1\times 2^1+0\times 2^0$$
$$=2^2+2^1+0$$

按十进制法则计算该多项式，

$$S=2^2+2^1+0=4+2+0=6$$

由以上的运算可以得出，一个二进制数转换为十进制数的法则为：把二进制数写成一个多项式，然后用十进制法则计算该多项式就可以得到二进制数所对应的十进制数。

3. 二化十例子

【例 8】 $1101_2=(?)_{10}$

$$S=1011_2=1\times 2^3+0\times 2^2+1\times 2^1+1\times 2^0$$
$$=2^3+0+2^1+1$$
$$=8+2+1$$
$$=11_{10}$$

【例 9】 $0.111_2=(?)_{10}$

$$S=0.111_2=1\times 2^{-1}+1\times 2^{-2}+1\times 2^{-3}$$
$$=2^{-1}+2^{-2}+2^{-3}$$
$$=0.5+0.25+0.125$$
$$=0.875_{10}$$

【例 10】 $111.11_2=(?)_{10}$

$$S=111.11_2=1\times 2^2+1\times 2^1+1\times 2^0+1\times 2^{-1}+1\times 2^{-2}$$
$$=2^2+2^1+2^0+2^{-1}+2^{-2}$$

$=4+2+1+0.5+0.25$

$=7.75_{10}$

由本例可以看出，当二进制数的权系数为0时，展成多项式时也使得该项为0，从而使多项式变得项数很少，转换起来十分简便。

【例11】 $1111111111_2=(?)_{10}$

$S=1111111111_2=2^9+2^8+2^7+2^6+2^5+2^4+2^3+2^2+2^1+2^0$

$=512+256+128+64+32+16+8+4+2+1$

$=1023_{10}$

在本例中由于二进制数的权系数都是1，从而使得多项式的计算量加大了。如果对二进制数进行某种变换，可以大大地减少计算量。

$S=1111111111_2$

$=1111111111_2+1-1$ （同时加上1，减去1）

$=10000000000_2-1=2^{10}-1=1024-1=1023_{10}$

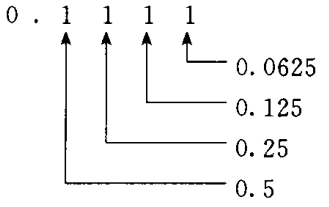
显然，通过这种变换可以加快二化十的速度。

【例12】 $0.1111_2=(?)_{10}$

$S=0.1111_2=1\times2^{-1}+1\times2^{-2}+1\times2^{-3}+1\times2^{-4}$

$=0.5+0.25+0.125+0.0625=0.9375_{10}$

由本例可以看出小数二化十的一个规律：



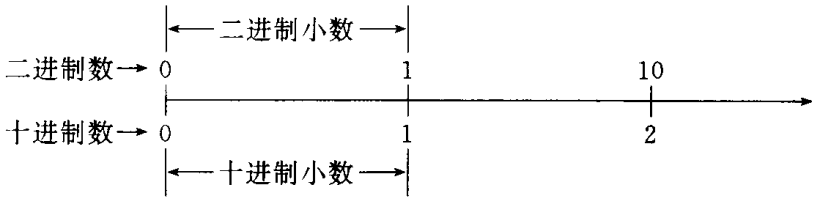
小数点后的第一个1为0.5；第二个1为0.25（即0.5的一半）；第三个1为0.125（即0.25的一半）；第四个1为0.0625（即0.125的一半），以此类推。根据这个规律，小数二化十时便可以速算。

【例13】 $0.11_2=(?)_{10}$

$S=0.11_2=0.5+0.25=0.75_{10}$

四、十进制数转换成二进制数

由二化十可以看出，任意一个二进制数必然对应一个十进制数。同理，一个十进制数也必然对应一个二进制数。为了进一步说明十进制与二进制关系，从而找到十进制数转换成二进制数的方法，先来观察下列数轴：



由数轴可以看出，十进制整数所对应的二进制数均为整数；十进制小数所对应的二进制数均

为小数。这种对应关系告诉我们，十进制数化为二进制数时，要把整数和小数分别进行转换。

1. 整数十化二的方法

对于任意的一个十进制整数，用 S 表示。它必然对应一个二进制整数，用 R 表示。则，

$$S=R \tag{1}$$

为简单起见，假定 R 是一个四位的二进制数 1111_2，我们可以用该数每一位的权做下标来表示这个二进制数：

$$\begin{array}{cccc} 1 & 1 & 1 & 1 \\ R_3 & R_2 & R_1 & R_0 \end{array}$$

于是，一个四位的二进制整数便可以用 $R_3\,R_2\,R_1\,R_0$ 这个序列来表示。即，

$$R=1111_2=R_3R_2R_1R_0$$

由（1）式可知 $S=R$，所以有：

$$S=R=1111_2=R_3R_2R_1R_0$$

即，

$$S=R_3R_2R_1R_0 \tag{2}$$

对（2）中的右端用多项式表示，则有：

$$S=R_3\times 2^3+R_2\times 2^2+R_1\times 2^1+R_0\times 2^0 \tag{3}$$

其中，$R_3=1$；$R_2=1$；$R_1=1$；$R_0=1$。

为了寻找整数十化二的方法，采取同时用 2 去除（3）式两端的方法，并观察所得余数的特征。

用 2 去除（3）式的两端，

$$\begin{aligned} S/2 &= R_3\times 2^3/2+R_2\times 2^2/2+R_1\times 2^1/2+R_0\times 2^0/2 \\ &= R_3\times 2^2+R_2\times 2^1+R_1\times 2^0+R_0 \end{aligned}$$

上式中，$R_0=1$，不能被 2 除尽，所以 R_0 为余数，而 $R_3\times 2^2+R_2\times 2^1+R_1\times 2^0$ 为商数。需要注意的是，余数 R_0 就是（2）式中二进制数的最低位。这说明对（3）式通过第一次除 2，就分离出二进制数的最低位。对于除以 2 所得到的商：

$$R_3\times 2^2+R_2\times 2^1+R_1\times 2^0$$

再用 2 去除则得，

$$\begin{array}{ll} R_3\times 2^1+R_2\times 2^0 & \text{余 } R_1 \\ \text{|←—商 数—→|} & \end{array}$$

而 R_1 为（2）式中的次低位。对于 $R_3\times 2^1+R_2\times 2^0$ 再用 2 去除得，

$$\begin{array}{ll} R_3\times 2^0 & \text{余 } R_2 \end{array}$$

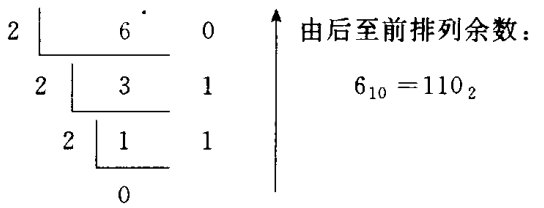
而 R_2 为（2）式中的次高位。对于 $R_3\times 2^0$ 再用 2 去除时，商为 0，余数为 R_3。而 R_3 为（2）式中的最高位。

通过以上操作，其结果是把十进制整数 S 所对应的二进制数 $R_3R_2R_1R_0$ 一一被分离出来，而被分离出来的二进制整数的每一位都是缩小了 2^0，2^1，2^2，2^3 后以余数的形式出现的。所以我们只要按照它们的位权排列成整数，就得到了十进制整数所对应的二进制整数。上面我们虽然只对一个四位二进制数进行了分析，但是，用这个特例所得出的规律是可以推广到任意位的。这个方法通常被称为“除 2 取余法”。

2. 整数十化二的操作

已知十进制整数，将其化为二进制整数所采用的方法是“除 2 取余法”。利用这一方法，对已知的十进制整数不断用 2 去除，直到商数为 0，将所得到的余数由后至前顺序排列为整数，即为十进制数所对应的二进制数。

【例 14】 $6_{10}=$ $(?)_2$

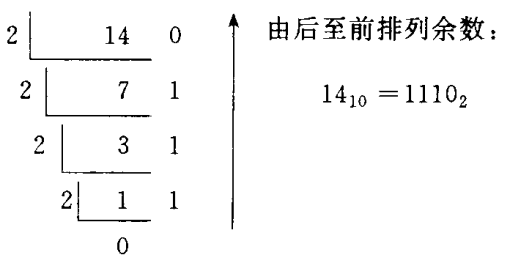


如果想验证十化二的结果是否正确，可以通过二化十的办法。例如，

$110_2=1\times2^2+1\times2^1+0\times2^0$

$=4+2+0=6_{10}$

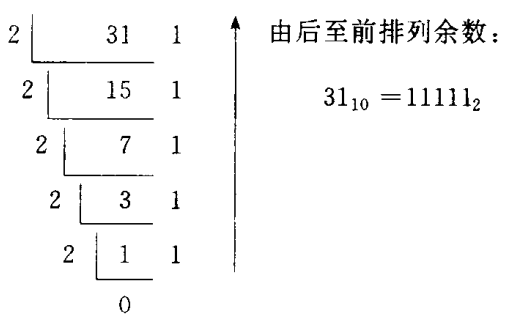
【例 15】 $14_{10}=$ $(?)_2$



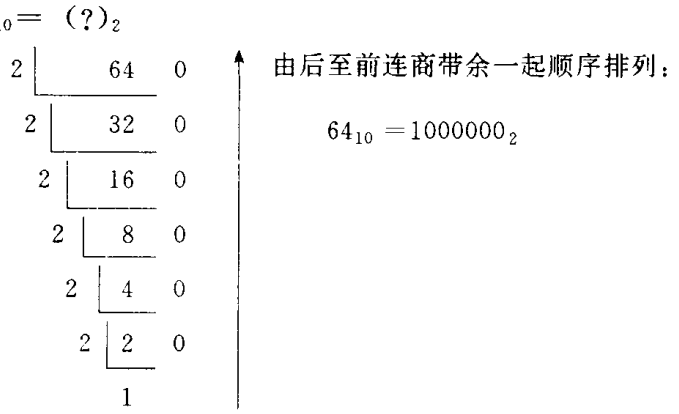
验证：$1110_2=1\times2^3+1\times2^2+1\times2^1+0\times2^0$

$=8+4+2+0=14_{10}$

【例 16】 $31_{10}=$ $(?)_2$



【例 17】 $64_{10}=$ $(?)_2$



由本例可以看出在整数十化二的操作中，当用 2 去除十进制整数时，只需要除到商数为 1 便