高等院校数学课程改革创新系列教材

计算机数学

◎ 游安军 编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY http://www.phei.com.cn

高等院校数学课程改革创新系列教材

计算机数学

游安军　编　著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 提 要

“计算机数学”是计算机类专业的一门基础课，它描述了计算机科学离散性的特点。全书共分6章，深入浅出地介绍了数字系统，集合、关系和函数，命题逻辑、谓词和量词、推理规则，算法基础（欧氏算法、递归算法等），图论，树和二叉树等基础知识。同时各章节配备了适量的习题供读者练习，以便读者切实掌握相应的数学知识，增强应用能力。

本书编排方式新颖，例题丰富详尽，语言通俗易懂，叙述清新自然，是计算机类专业数学基础的极佳入门教材，尤其体现了数学课为计算机技术人才培养服务的理念，特别适合作为高等学校应用型本科和高职高专计算机类专业的数学教材，也可供计算机领域的其他工程技术人员阅读和参考。

图书在版编目（CIP）数据

计算机数学/游安军编著.—北京:电子工业出版社，2013.9
高等院校数学课程改革创新系列教材
ISBN 978-7-121-21429-5

Ⅰ.①计…　Ⅱ.①游…　Ⅲ.①电子计算机-数学基础-高等学校-教材　Ⅳ.①TP301.6

中国版本图书馆CIP数据核字(2013)第212284号

策划编辑：朱怀永
责任编辑：朱怀永
印　　刷：三河市鑫金马印装有限公司
装　　订：三河市鑫金马印装有限公司
出版发行：电子工业出版社
　　　　　北京市海淀区万寿路173信箱　邮编100036
开　　本：787×1092　1/16　印张：16.25　字数：416千字
印　　次：2013年9月第1次印刷
定　　价：32.80元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010)88254888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：(010)88258888。

前　言

作为计算机数学的入门教材，本书主要介绍计算机技术、信息科学中所涉及的最基础的数学知识。其目的是不仅展示数学的应用性，而且通过数学知识的扩展来培养学生的逻辑思维和抽象能力，为计算机专业学生的可持续发展建立必要的基础。

编写说明

20世纪著名数学家，德国哥廷根学派的重要成员 R. 柯朗在其名著《What is Mathematics》(1941年)中说，“两千多年来，人们一直认为每一个受教育者都必须具备一定的数学知识。但是今天，数学教育的传统地位却陷入了严重的危机之中，而且遗憾的是，数学工作者要对此负一定的责任。……数学研究已经出现一种过分专门化和过于强调抽象的趋势，而忽视了数学的应用以及与其他领域的联系，不过，这种状况丝毫不能证明紧缩数学教育的政策是合理的。相反，那些醒悟到培养思维能力的重要性的人，必然会采取完全不同的做法，即更加重视和加强数学教学。”转眼70多年过去了，数学教育中“忽视了数学的应用以及与其他领域的联系”的状况仍然没有实质性的改观，我国学校教育尤其如此。近十多年来，随着我国高等教育的大众化和普及化，尤其是高等职业教育的迅速发展，这种状况已经到了迫不得已必须要变革的时候了。也就是说，大学数学课程再也不能仅仅停留在“培养思维能力”的层面，还必须为学生的专业知识学习和能力发展提供实在的支持。鉴于这种考虑，很长时间以来我们一直都在思考计算机专业的数学课程建设问题。为此，我们认真学习了国内许多学者编写的《计算机数学》教材，虽然受到一些启发，但还是觉得不满意。这主要表现在：一是内容选取不恰当，没有体现出为计算机专业服务的个性特点；二是沿袭着比较传统的学科体系的编写思路。于是，我们把眼睛转向了国外比较流行的相关教材，通过不断地学习、比较和实践，我们采取了与国内诸多教材完全不同的编写思路和内容设计。

这门课程的内容可以有不同选择。国内多个版本的《计算机数学》教材都包含着传统的微积分，或者个别教材使用数学软件 Mathematica 或 Matlab 进行编程实践。我们认为，对计算机专业来说，这些内容并不是最重要的，其实用价值也不是最高的。于是，我们放弃了比较传统的微积分和线性代数等内容，而把离散数学作为计算机数学的主体。它包括数的进制，集合、关系和函数，命题逻辑和量词，算法基础，图论、树和二叉树等基础知识。这与大家已经见到的、由国内学者编写的《计算机数学》有非常大的差异。

同时，那些教材基本上没有算法实践，而本教材比较重视编程和算法设计。算法是计算机数学应用的重要方面，教材中的算法程序(段)和过程都以伪代码的形式给出。这样做能更好地把数学知识与计算机应用融为一体，通过接触不同的算法来培养学生的编

程能力，从而体现作为专业基础的数学课程直接为计算机专业服务的理念。

再者，本教材所列出的主题有自己特别的安排顺序。经验告诉我们，过早地介绍命题与逻辑，效果可能并不理想。考虑到学生的数学基础，先讲解集合、关系和函数等相对比较熟悉的内容可能是恰当的；然后再对逻辑给出清晰的阐述。而所有的不同都体现了我们对这门课程的独特理解。

尽管我们已经付出了十分艰辛的努力，但本教材仍然会存在诸多不足，尤其是在内容选取和编排方式上。如果读者朋友有任何的改进教材的建议，请发送邮件到 anjun65@sina.com，我们会非常认真地听取并做出积极的响应。

致学生

对于刚刚结束高中阶段学习进入到大学计算机专业的学生来说，也许并不喜欢数学，但在大学第一年的学习中，数学是重要的必不可少的基础课程。计算机及其相关领域里的许多工作都与编程或编程思想有着千丝万缕的联系，而其中但凡有点创造性的工作都是基于编程的。这些工作对逻辑思维和抽象能力有比较高的要求，学习数学是达到这种要求的重要途径。因此，我们建议读者根据“取法其上，得乎其中；取法其中，得乎其下；取法其下，则法不得也”的思路，尽自己的最大努力多学一些数学知识，给自己预留多一点发展空间。尽管这样做有时候是比较辛苦的，而一旦掌握了数学知识，你就会发现，在计算机领域之外，这些知识也大有裨益。

阅读本书基本上不需要微积分和计算机方面的知识预备。但是，如果你已经掌握了基础水平的高中数学，或曾经历过程序设计方面的练习，读起来会更加轻松自如。考虑到同学们以前的数学基础和学习兴趣，我们花费了相当大的精力尽量把数学知识讲得更加的浅显易懂，但初学者可能还是会感到有点不适应。此时你需要沉静和一些耐心，以及咬住青山不放松的勇气。人类历史上最伟大的数学家牛顿和高斯之所以能在数学上做出如此多的贡献，不仅因为他们天资聪慧，还因为他们在伟大工作上的长期准备和不间断地思考。因此，只要我们不间断地思考，就应该相信自己在老师和周围同学的帮助下能够克服暂时的困难，取得实质性的数学进步。

数学不像其他学科，如历史学、医药学、法律等，有大量的内容需要记忆。数学学习最重要的是理解和运用，而理解与运用总是相互依存的。为了获得理解，经常需要从一些特例出发，再推及较大或更广的范围，从而达到抽象的层次；或者是反复考察数学定义所涉及的正面和反面的例证。读数学书时一定要养成用实例论证的习惯。因此，本书包含了极其丰富的例题和适量的习题，它们是每一章节的重要组成部分。解决这些问题是你理解数学知识和扩展抽象能力的必要环节。因为无论一个概念看上去是多么的简单，如果不通过实例的演练，你根本无法完全掌握。尽可能多地做练习，你才能从本书中获得最大的益处。

虽然本书的附录提供了许多习题的解答，但我们还是期待读者能够通过自己的努力解决它们，因为解题所需要的抽象力、理解力是其他任何人都无法代为传授的，它们是在

"做"与"思"的过程中所达到的对数学知识的升华。

致教师

对于长期讲授连续性数学(比如微积分)的教师来说,本书中的个别内容或许有点陌生。但是,所有的这些内容对计算机专业来说都是基础的和重要的。尽管我们习惯于在自己熟悉的知识领域里工作,计算机专业的数学教师还是应该在课程内容的取向上做出积极的改变。就内容的针对性而言,《计算机数学》比传统的《高等数学》更能为计算机技术课程提供必要的知识基础,所以我们尝试着做出这种改变。虽然我国高等职业院校已经有学者编写并出版了类似名称的教材,但如此巨大的改变在已经过去的十多年时间里几乎没有存在过。而把这门课程付诸实践,必然会调整数学教师的知识结构和教学方法。这一切都需要相当的勇气和持之以恒的努力。

关于本教材,建议的学时数约为72个左右,各个学校可根据自己的实际情况适当安排。同时,我们建议本课程与"C语言程序设计"课程同期进行教学,而且担任本课程教学的数学教师懂一点C语言知识。我们推荐师生协同的小组合作学习模式,让学生充分地参与到教学过程之中讨论学习内容,并自主完成相应的练习,这样做的效果会比较好。不要低估学生的自主学习能力,相反,要改变的是自己的教学观念、教学设计和课堂管理。而算法部分的许多内容还需要老师组织学生写出相应的C程序,并上机实践。

致 谢

首先感谢国内外的许多学者,美国的 Richard Johnsonbaugh, David Makinson, Kenneth H. Rosen 及中国的邓辉文等。他们的著作给我们提供了重要材料和写作思路。同时也非常感谢珠海城市职业技术学院工程与信息学院院长刘辉珞教授。他不仅深深地知道数学在技术性学科和经济管理中的重要性和应用性,而且对高等职业教育数学课程改革的难度有充分的估计。因此,多年来他一直鼓励我们要抓住高等职业发展教育的契机,从课程和教材建设入手,真正地行动起来,在高等职业教育领域的数学课程建设与教学改革方面做出一些有影响的创新性工作。本书的写作和出版承载着他厚厚的期待。

感谢珠海城市职业技术学院工程与信息学院计算机教研室的所有同事,从与这个集体里众多优秀老师的交流和讨论中,我们获得了许多有益的意见。珠海城市职业技术学院李平老师和人文社科系2009级胡晓珠、陈颖和叶聪玲等同学为本书的写作付出了大量的时间和精力,计算机应用技术专业2011级詹宏棍、曾碧芳等同学也为本书做了得力工作。珠海城市职业技术学院图书信息中心流通阅览部王海英主任为本书提供了重要的参考资料。珠海市第九中学的黎华女士耐心地绘制了本书中绝大多数的图表。在此一并致谢。

也要真诚地感谢帮助自己成长的许多朋友,而以下诸位是不得不提到的。他们是珠海城市职业技术学院原教务处处长张建夕副教授、工程与信息学院陈国康副教授、人文社科系方守金教授、艺术设计系林跃明教授,以及沈国辉老师等。自1993年从湖北大学

数学系硕士生研究生毕业到今天，我还能坚持自己的信念，忠于自己的土壤，尽自己的能力做一些有益于学生和学术的事情，这都得益于诸多朋友的提醒、督促和帮助。感谢他们坚定了我的发展方向。

最后要感谢电子工业出版社束传政和朱怀永先生，正是因为他们对高等职业教育数学课程与教材建设有着独到的、前瞻性的认识，才使得本书最终能与广大读者见面。

著　者

2013 年 6 月于珠海

目　录

第 0 章　数字系统

计算机是一个逻辑运算机器。它借助于电子线路的 on(开)和 off(关)两种状态的组合来表达信息,这两种状态在数学里被解释为 1 和 0。而其他的数、文字或字符都要通过这两个数字进行相应的编码,才能被计算机理解。因此,以 0 和 1 为代码的二进制是计算机技术的数学基础。本章主要介绍二进制计数系统和位的简单知识。

0.1　数的进制

十进制是我们熟悉的计数系统,它用十个符号 0,1,2,3,4,5,6,7,8,9 来表示整数。在表示一个整数时,符号的位置(也就是我们过去常说的位置值原则)是重要的:从右边开始,第一个符号表示 1($1=10^0$)的个数,第二个符号表示 10($10=10^1$)的个数,第三个符号是 100($100=10^2$)的个数,…,第 n 个符号代表 10^{n-1} 的个数,等等。例如,$73854=7\times10^4+3\times10^3+8\times10^2+5\times10^1+4\times10^0$。我们把系统所基于的那个数(十进制系统就是 10)称为基数。

计算机科学要用到 10 以外的基数。比如,二进制就是以 2 为基数的计数系统。下面我们讨论二进制系统。

在二进制数字系统(基数为 2)中,只用两个符号 0 和 1 来表示一个整数。从右边开始,第一个符号表示 1($1=2^0$)的个数,第二个符号表示 2($2=2^1$)的个数,第三个符号是 4($4=2^2$)的个数,第四个表示 8($8=2^3$)的个数,…,第 n 个位置的符号表示 2^{n-1} 的个数。比如,二进制数 101011111_2 可以化为十进制数

$$1\cdot2^8+0\cdot2^7+1\cdot2^6+0\cdot2^5+1\cdot2^4+1\cdot2^3+1\cdot2^2+1\cdot2^1+1\cdot2^0=351$$

如果不知道使用的是什么数制系统,一个数字所表示的意义将是含混的。比如,101101 在十进制中表示某个数,在二进制中则表示完全不同的另一个数。通常情况下,读者能从上下文知道所用的是什么数制。但是,如果我们要完全地明确表示一个数,就把基数作为下标来标记所用的数制系统——十进制用下标 10,二进制用下标 2,等等。例如,二进制的 101101 用 101101_2 来表示,十进制的 101101 用 101101_{10}来表示。

例 0.1.1

二进制数 101101_2 表示此数由一个 1、没有 2、一个 4、一个 8、没有 16、一个 32 等组成。它可以表示为

$$101101_2=1\cdot2^5+0\cdot2^4+1\cdot2^3+1\cdot2^2+0\cdot2^1+1\cdot2^0$$

用十进制计算上面式子的右边,得出

$$
\begin{aligned}
101101_2 &= 1\cdot 2^5+0\cdot 2^4+1\cdot 2^3+1\cdot 2^2+0\cdot 2^1+1\cdot 2^0 \\
&= 32+8+4+1 \\
&= 45_{10}
\end{aligned}
$$

此例告诉我们，如何把一个二进制数转换为一个十进制数。

下面考虑相反的问题——把一个十进制数转换为一个二进制数，例如，要把十进制数 91 转换为二进制数。首先我们用 91 除以 2，得到

$$91=2\cdot 45+1 \tag{0.1.1}$$

再把 45 除以 2，得

$$45=2\cdot 22+1 \tag{0.1.2}$$

把上式的 45 代入式(0.1.1)，得

$$
\begin{aligned}
91 &= 2\cdot 45+1 \\
&= 2\cdot(2\cdot 22+1)+1 \\
&= 2^2\cdot 22+2+1 \qquad (0.1.3)
\end{aligned}
$$

再把 22 除以 2，得

$$22=2\cdot 11$$

将它代入式(0.1.3)，得

$$
\begin{aligned}
91 &= 2^2\cdot 22+2+1 \\
&= 2^2\cdot(2\cdot 11)+2+1 \\
&= 2^3\cdot 11+2+1 \qquad (0.1.4)
\end{aligned}
$$

再把 11 除以 2，得

$$11=2\cdot 5+1$$

将它代入式(0.1.4)，得

$$91=2^4\cdot 5+2^3+2+1 \tag{0.1.5}$$

再把 5 除以 2，得

$$5=2\cdot 2+1$$

将它代入式(0.1.5)，得

$$
\begin{aligned}
91_{10} &= 2^5\cdot 2+2^4+2^3+2+1 \\
&= 2^6+0+2^4+2^3+0+2+1 \\
&= 1011011_2
\end{aligned}
$$

上面的计算表明，当 N(或商)逐次被 2 除时所得的余数，就给出了 N 的二进制表示的各个位置上的数。在(0.1.1)式中，第一次被 2 除的余数就是最低位(1 的个数)，在(0.1.2)式中，第二次被 2 除的余数就是代表 2 的个数的位，等等。

例 0.1.2

把十进制 131 写为二进制。

下面的计算表明，逐次被 2 除，余数就是从右边开始的各个二进制位。

$$
\begin{array}{llll}
2) & \underline{131} & 余数=1 & 2^0 的位 \\
2) & \underline{65} & 余数=1 & 2^1 的位 \\
2) & \underline{32} & 余数=0 & 2^2 的位 \\
2) & \underline{16} & 余数=0 & 2^3 的位 \\
2) & \underline{8} & 余数=0 & 2^4 的位 \\
2) & \underline{4} & 余数=0 & 2^5 的位 \\
2) & \underline{2} & 余数=0 & 2^6 的位 \\
2) & \underline{1} & 余数=1 & 2^7 的位 \\
 & 0 & &
\end{array}
$$

直到所除得的商数为 0 时，停止计算。第一次被 2 除的余数就是最低位（1 的个数），第二次被 2 除的余数就是表示 2 的个数的位，等等。也就是说，将所得余数从下往上排列为从左向右的形式，我们就得出：

$$131_{10}=10000011_2$$

十进制数相加的方法也可以用于二进制数相加，但是，我们要把十进制加法表用二进制加法表代替，见表 0.1.1。

表 0.1.1

+	0	1
0	0	1
1	1	10

在十进制中，$1+1=2$，且 $2_{10}=10_2$。所以，在二进制中，$1+1=10$。

例 0.1.3

把二进制数 10011011 和 1001011 相加。

我们把问题写为

$$
\begin{array}{r}
10011011 \\
+\quad 1001011
\end{array}
$$

与十进制加法一样，我们从右边开始，把 1 和 1 相加，其和为 10_2，于是我们写出 0 并有进位 1。此时计算成为

$$
\begin{array}{r}
\mathit{1} \\
10011011 \\
+\quad 1001011 \\
\hline
0
\end{array}
$$

下面，把三个 1 相加，得 11_2 写出 1，还有进位 1。此时，计算成为

$$
\begin{array}{r}
\mathit{1} \\
10011011 \\
+\quad 1001011 \\
\hline
10
\end{array}
$$

继续此法，我们得出

$$\begin{array}{r} 1001\ 1011 \\ +\quad 100\ 1011 \\ \hline 1110\ 0110 \end{array}$$

在计算机系统中，重要的数制还有八进制和十六进制的系统。下面我们讨论十六进制系统，而把八进制系统作为练习题。

在十六进制系统中，表示整数的符号是：0，1，2，3，4，5，6，7，8，9，A，B，C，D，E 和 F 共 16 个字符，其中的符号 A～F 代表十进制数的 10～15（一般来说，N 进制的系统，需要 N 个不同的符号来表示 0，1，2，…，$N-1$）。在表示一个整数时，从右边开始，第一个符号表示 1 的个数，第二个符号表示 16 的个数，第三个符号表示 16^2 的个数，…，第 n 个位置的符号表示 16^{n-1} 的个数，等等。

例 0.1.4

把十六进制数 B4E 转换为十进制数。

我们有

$$\begin{aligned} \mathrm{B4E}_{16} &= 11 \cdot 16^2 + 4 \cdot 16^1 + 14 \cdot 16^0 \\ &= 11 \cdot 256 + 4 \cdot 16 + 14 \\ &= 2816 + 64 + 14 \\ &= 2894_{10} \end{aligned}$$

类似于二进制的讨论，要把一个十进制数转换为十六进制数，我们逐次把它（或者商）除以 16，所得的余数就给出了十六进制的各个数位上的符号。

例 0.1.5

把十进制数 20385 转换为十六进制数。

用 20385（或商）依次除以 16，余数就是从右边开始的各个十六进制位上的符号。

16)20385	余数＝1	16^0 的位
16)1274	余数＝10＝A	16^1 的位
16)79	余数＝15＝F	16^2 的位
16)4	余数＝4	16^3 的位
0		

一直到商数为 0 时，停止计算。第一次被 16 除的余数就是最低位（1 的个数），第二次被 16 除的余数就是 16 的位置上的数，如此等等，我们得出：

$$20385_{10} = 4\mathrm{FA1}_{16}$$

下面的例子表明，我们可以用十进制数相加的方法来进行十六进制数的相加。

例 0.1.6

把十六进制数 84A 和 42EF 相加。

问题可以写为

$$\begin{array}{r} 84A \\ +\ 42EF \\ \hline \end{array}$$

我们从最右边开始，把 F 和 A 相加。由于 F 是 15_{10}，A 是 10_{10}，所以

$$\begin{aligned} F + A &= 15_{10} + 10_{10} \\ &= 25_{10} \\ &= 19_{16}\text{（满 16 进 1，写出余下的 9）} \end{aligned}$$

$$\begin{array}{r} \mathit{1} \\ 84A \\ +\ 42EF \\ \hline 9 \end{array}$$

下面我们把 1，4 和 E 相加，满 16 进 1，写出余下的 3，得 13_{16}。

$$\begin{array}{r} \mathit{1} \\ 84A \\ +\ 42EF \\ \hline 39 \end{array}$$

继续照此进行，得到

$$\begin{array}{r} 84A \\ +\ 42EF \\ \hline 4B39 \end{array} \quad \text{对应的十进制加法是} \quad \begin{array}{r} 2122 \\ +\ 17135 \\ \hline 19257 \end{array}$$

例 0.1.7

将二进制数转换成八进制数或十六进制数的方法是：从最低位开始向左边按每 3 位（转换成八进制数）或每 4 位（转换成十六进制数）分组，最后不满 3 位或 4 位的，则填 0 补充，再将每组用对应的八进制数或十六进制数代替，即可得相应的八进制数或十六进制数。

把二进制 1110110100 转换成八进制。

二进制	1,110,110,100	每 3 位一组
	001,110,110,100	最高位补 0
八进制	1　6　6　4	结果

于是，$1110110100_2 = 1664_8$

把二进制 1110110100 转换成十六进制。

二进制	11,1011,0100	每 4 位一组
	0011,1011,0100	最高位补 0
16 进制	3　B　4	结果

于是，$1110110100_2 = 3B4_{16}$

例 0.1.8

将八进制数或十六进制数转换成二进制数的方法是：将八进制数和十六进制数的每一位用对应的 3 位或 4 位二进制数来表示即可(参见表 0.1.2)。

把八进制 327 转换成二进制。

八进制　　3　　2　　7

二进制　　011　010　111

于是，$327_8=011010111_2$

把十六进制 7A 转换成二进制。

十六进制　　7　　　A

二进制　　0111　　1010

于是，$7A_{16}=1111010_2$

表 0.1.2

十进制	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
十六进制	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
八进制	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
二进制	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111

习题 0.1

1. 把下列各二进制数改写为十进制数和十六进制数。

(1) 100101　(2) 101011　(3) 10011011　(4) 1100000

2. 把各十进制数表示为二进制数和十六进制数。

(1) 34　(2) 61　(3) 403　(4) 1024

3. 把下列二进制数相加。

(1) 1011+1110　(2) 101011+1101

(3) 110110+101101　(4) 101101+110101

4. 把十六进制数表示为十进制数和二进制数。

(1) 3A　(2) 1E9　(3) 3F7C　(4) A03

5. 把下列各十六进制数相加。

(1) 4A+B4　(2) 49F7+C6E　(3) 3B9C+9E2D

6. 2010 是一个二进制数吗？是一个十进制数吗？是一个十六进制数吗？

7. 1001010 是一个二进制数吗？是一个十进制数吗？是一个十六进制数吗？

8. 完成表 0.1.3 所列的八进制加法表。

表 0.1.3

+	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

9. 把下列八进制数转化为十进制数、二进制数、十六进制数。

(1) 73　　(2) 7654　　(3) 632　　(4) 207

0.2　位的知识

计算机中的数据和信息都要表示为 1 和 0 的组合。在这种组合的表达式中，我们把每 8 个二进制位合称为一个字节。比如，1101 0101 就是一个字节，0100 0001 1001 1101 就是两个字节，其中的每一个二进制位称为一个比特。我们可以从左至右给一个字节的 8 个位排序（依次记为从 7 至 0）。现在想象有这么一个字节，如图 0.2.1 所示。

	(最高位)							(最低位)
字位数	7	6	5	4	3	2	1	0
	0	1	0	0	1	0	0	1
位置值	128	64	32	16	8	4	2	1

图 0.2.1

在上图中，$128=2^7$，$64=2^6$，…，依此类推。因此，一个字节所能表示的最大数是各位均被置为 1 的数：1111 1111。该二进制数的值等于：

$$128+64+32+16+8+4+2+1=255$$

最小的二进制数是 0000 0000。所以，一个字节能存放从 0 到 255 这个范围内的数，总共有 256 个可能的值。

大多数计算系统的内存储器是由许许多多被称为字节的单元组成的。每一个字节有一个地址。每一个存储单元存放一个数据或一条指令。在微型机中一般以 4 个字节存放一个实数，以 2 个字节存放一个整数。为简化起见，下面我们只用一个字节存放一个整数。

如果存放的是带符号的数，通常将第 7 个字位放置符号，非负整数置为 0，负整数置为 1。比如，

+7　的原码为　0000 0111（最左边的 0 表示非负整数）

−7　的原码为　1000 0111（最左边的 1 表示负数）

二进制数 111 代表十进制的 7。如果用 2 个字节存放一个整数，情况是一样的。比如 +7 的原码为 0000 0000 0000 0111。

根据以上规则，0 的原码如下：

+0 的原码为 0000 0000

−0 的原码为 1000 0000

这样一来,表示同一个数 0 的+0 和−0 在内存中就有两个不同的形式,也就是说 0 的表示不是唯一的。这显然不适合于计算机的运算。下面介绍反码。

如果一个数值为正,则它的反码与原码相同。比如,

+7 的反码为 0000 0111

127 的反码为 0111 1111

+0 的反码为 0000 0000

如果一个数值为负,则其符号位仍为 1,其余各位是对原码取反(也就是把 0 换为 1,1 换为 0 所得)。比如,

−7 的反码为 1111 1000

−127 的反码为 1000 0000

−0 的反码为 1111 1111

此时,0 的表示仍是不唯一的。

原码和反码都不便于计算机的运算,因为在运算过程中要先判断各自的符号位,然后才能对后 7 位进行相应的处理,很不方便。为了将符号位和其他位进行统一处理,对减法也能按照加法来处理,这就需要补码。补码是这样规定的。

如果一个数是正数,其原码、反码、补码都相同。比如,

+7 的补码为 0000 0111

如果一个数是负数,其补码的最高位仍为 1,其余各位为原码的取反,然后对整个数再加 1。比如,

−7 的补码为 1111 1001

−0 的补码为 0000 0000

+0 的补码为 0000 0000

此时,+0 和−0 的补码表示是相同的,也就是说,0 的补码是唯一的。十进制数与补码的对应关系见表 0.2.1。

表 0.2.1

数 值	补 码
0	0000 0000
−1	1111 1111
−2	1111 1110
…	…(往下不断减 1)
−127	1000 0001
−128	1000 0000
1	0000 0001
2	0000 0010
…	…(往下不断加 1)
126	0111 1110
127	0111 1111