普通高等教育“十二五”规划教材
电工电子基础课程规划教材

微机原理与接口技术
（第2版）

■ 黄玉清　刘双虎　杨胜波　主编

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

普通高等教育“十二五”规划教材
电工电子基础课程规划教材

微机原理与接口技术

（第2版）

黄玉清　刘双虎　杨胜波　主编

胡　捷　谭顺华　陈春梅　唐东明　编

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

内 容 简 介

本书是省精品课程、省“质量工程”精品教材，依据电工电子基础平台课程教学基本要求编写，以8086CPU作为教学芯片，介绍微机原理的基本理论与技术应用。主要内容包括：绪论、8086微处理器、8086指令系统、汇编语言程序设计、微机的输入与输出、中断系统、定时/计数技术、并行接口、串行通信接口、DMA控制器、存储器、数/模和模/数转换、课程综合设计等。本书提供大量实例，配套电子课件、习题参考答案和课程教学网站等。

本书可作为高等学校电子信息与电气专业和计算机专业相关课程的教材，也可作为培训、自学和考研的参考用书，还可供相关领域的科技工作者学习参考。

图书在版编目（CIP）数据

微机原理与接口技术 / 黄玉清，刘双虎，杨胜波主编. —2版. — 北京：电子工业出版社，2015.8
电工电子基础课程规划教材
ISBN 978-7-121-26340-8

I. ①微… II. ①黄… ②刘… ③杨… III. ①微型计算机－理论－高等学校－教材 ②微型计算机－接口－高等学校－教材 IV. ①TP36

中国版本图书馆CIP数据核字（2015）第130318号

策划编辑：王羽佳
责任编辑：王晓庆
印　　刷：北京中新伟业印刷有限公司
装　　订：北京中新伟业印刷有限公司
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：18.5　字数：534千字
版　　次：2011年6月第1版
　　2015年8月第2版
印　　次：2015年8月第1次印刷
印　　数：3000册　　定价：39.90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。

第 2 版前言

本书是省精品课程、省“质量工程”精品教材。

随着微型计算机技术的迅猛发展，为适应本科教育教材更新需求和电子信息科学与电气信息类专业课程基本要求，根据作者多年来从事高校“微机原理及应用”、“单片机原理及应用”课程的教学实践和科研开发的切身经验，以培养学生计算机设计应用系统能力的目的，并考虑近年来嵌入式系统、微电子和通信等技术领域的迅速发展和需求，我们组织编写了本书。

本书详细地从概念上讲述了计算机的基本组成和工作原理，特别是用简单模型机形象、直观地介绍了计算机的主要工作过程，使学生对计算机的原理和运行机制有较深刻的理解。

第 2 版仍然保持原来的写作风格，但在内容的编排顺序上，调整个别章节，删除了 80x86 一章，并进行了仔细校对。由于微机原理与接口技术作为非计算机专业的硬件技术基础，举足轻重。随着微处理器技术的发展，32 位微处理器的使用相当普遍。但用 32 位 CPU 作为初学者的教学芯片，所需基础知识量大，还需进一步探索。当前用 8086 CPU 作为教学芯片来介绍微机组成，我们认为还是恰当的。因此本书主要以 8086 CPU 为例，重点介绍微机组成原理与结构，调整了汇编语言与设计。为保持本书知识体系的完备，本书与前一版的核心内容基本一致。

本书可作为高等学校电子信息与电气专业和计算机专业“微机原理”、“微机原理与接口技术”等课程的教材。本书面向普通学生，入门要求降低。本书充分考虑到普通高等学校本、专科学生及自学人员的实际知识水平，以清晰的逻辑结构展开教学内容，尽量使用浅显生动的语言，不惜笔墨详尽讲解重点和难点知识。本书可被用于不同对象、不同层次、不同课时的教学。

本书提供配套电子课件、习题参考答案和课程教学网站，请登录华信教育资源网（http://www.hxedu.com.cn）注册下载。

本书第 1、2、8 章由西南科技大学杨胜波副教授编写，第 3、6、7 章由西南科技大学黄玉清教授编写，第 4 章由西南科技大学谭顺华副教授编写，第 10、11 章由成都信息工程大学刘双虎编写，第 5、12 章由西南科技大学胡捷副教授编写，第 9 章由西南科技大学唐东明博士编写，第 13 章与附录由西南科技大学陈春梅副教授编写，全书由黄玉清教授主编并统稿。

本书得到四川省精品课程、四川省高等教育“质量工程”之精品教材建设、西南科技大学精品课程（031222）、西南科技大学教材规划项目（06jc0027）资助。

本书是作者在多年积累的相关教学实践和科研成果的基础上编写而成的。西南科技大学吴坚教授、陈波教授（博士）、李磊民教授、江虹教授（博士）对本书提出了诸多宝贵的修改意见，在此表示衷心的感谢！本书的编写过程中，参考了大量国、内外文献资料，在此，特向有关作者表示衷心的感谢！

限于作者的水平，且时间有限，缺点和错误在所难免，殷切期望广大读者提出宝贵意见，敬请批评指正。

作　者

2015 年 7 月

目　录

第1章 绪 论

微机（微型计算机）是大规模、超大规模集成电路的产物，具有成本低、性能强的特点，是目前应用最广泛的计算机种类。微机具有独特的组成结构，但在实现原理和运行机制上与通用计算机没有本质区别。

本章首先简要介绍计算机的发展历程，接着以数字计算机的信息编码入手，讨论计算机的组成结构、微机的结构特点，通过一个模型机说明数字计算机的运行原理，引出总线概念和时序概念，最后简要介绍微机系统的三个层次，并给出本课程学习的一个实际 PC 微机系统结构。

本章为后续各章提供必要的概念基础。建议本章学时为 2～3 学时。

1.1 计算机发展概述

1.1.1 早期计算机

人类使用进位制计数系统以来，不同地区曾相继出现过各种计算工具。已知最早的计算工具是中国古代的算筹，后来演变出算盘，古巴比伦、古埃及、古希腊、古印度、古罗马等文明古国也都先后出现了各自的计算工具，如算盘、算板、沙盘等。这些计算工具的原理基本相似，都是使用某种具体的物体来代表数，人们按规则操作这些物体来实现计算，实际上计算过程仍然是由人脑完成的，这些物体只起到暂时存储中间结果的作用。近代，欧洲出现了比例规、对数计算尺等计算工具，进一步简化了操作过程，但仍然脱离不了人脑的参与。人类随着对未知领域认识的不断拓展，对于计算精度和计算速度的要求越来越高。一直以来，人类都在尝试制造能够代替人脑计算的更为高效、更为可靠的工具。

1623 年，德国人威廉·施卡德（Wilhelm Schickard）制造出第一部机械计算器，这台机器采用钟表齿轮原理，能进行 6 位数的加减运算，通过钟声输出结果。施卡德的机器在计算工具上的突破意义远远超出了其实用价值，它在无人参与的情况下，第一次自动完成了计算过程。在这之后，类似的发条动力、手摇动力机械计算器不断涌现。1642 年，历史上著名的法国科学家布莱士·帕斯卡（Blaise Pascal）为税收统计发明了一种称为 Pascaline 的轮式计算器（Wheel Calculator），可进行 6 位十进制数、一位二十进制数和一位十二进制数（与法国当时的货币制相应）的加法和减法运算。1673 年，德国数学家哥弗雷德·莱布尼茨（Gottfried Leibniz）在 Pascaline 基础上增加了阶梯式圆柱齿轮，制造出可进行四则运算的步进计算器（Stepped Reckoner）。

1801 年，法国人约瑟夫·马利·杰卡德（Joseph Marie Jacquard）发明了一种提花织布机，可以通过穿孔卡（Punched Card）来设定织花图样，是第一种可编程性质的机器。

1832 年，英国天文学家查尔斯·巴贝奇（Charles Babbage）开始着手设计制造分析机（Analyse Engine）。分析机以蒸汽机为动力，主要包括三部分：Store 的齿轮阵列，可存储 1000 个 50 位数；Mill 的齿轮计算装置，类似帕斯卡轮式计算器，可进行 50 位数的运算；第三部分没有具体命名，受杰卡德织布机的启发，以穿孔卡控制整个机器的操作顺序。除此之外，分析机还设计有在 Store 和 Mill 之间往返传送数据的部件，以及输出结果的打印部分，这种结构与现代通用计算机已经非常接近了。分析机受限于当时的机械加工精度，并没有全部完成，但曾经成功运行了用穿孔卡描述的伯努利

（Bernoulli）数计算程序，是第一台自动计算和可编程的机器，其设计思想为现代通用计算机奠定了理论基础。

进入 20 世纪以后，数学和工业技术都得到了充分的发展，数学的成就为计算机的设计提供了理论基础，而工业技术的改进为计算机的制造提供了物质基础。

1936 年，英国数学家阿兰·图灵（Alan Turing）发表了论文《论可计算数在判定问题中的应用》（*On Computer Numbers with an Application to the Entscheidungs Problem*），以布尔代数为基础，从理论上证明了可将逻辑中的任意命题用一种通用的机器来表示和完成，这种抽象的机器模型被称为图灵机（Turing Machine），是最早提出的通用计算机定义，并成为计算理论的核心思想。

1937 年，美国数学家克劳德·香农（Claude Shannon）发表了论文《继电器和开关电路的符号分析》（*A Symbolic Analysis of Relay and Switching Circuits*），建立了使用开关电路实现逻辑和数学运算的理论基础。

1941 年，德国人康拉德·楚泽（Konrad Zuse）采用继电器制造出第一台符合图灵机定义的通用计算机 Z-3。Z-3 采用二进制计数，程序输入采用穿孔电影胶片，数据输入由一个数字键盘输入，计算结果用小电灯泡显示。1944 年，美国数学家霍德华·艾肯（Howard Aiken）根据巴贝奇分析机的设计思想，也制造出一台继电器计算机 ASCC（Automatic Sequence Controlled Calculator，全自动化循序控制计算机），其用户哈佛大学将其命名为 Mark Ⅰ。Mark Ⅰ采用十进制计数，数据和指令通过穿孔卡片机输入，输出则通过电传打字机，由它计算的《数学用表》至今还在使用。

1939 年，美国爱荷华州立大学的约翰·文森特·阿塔纳索夫（John Vincent Atanasoff）及研究生克里福特·贝瑞（Clifford Berry）开发出第一台电子计算机，被称为阿塔纳索夫-贝瑞计算机（Atanasoff-Berry Computer）。这台计算机使用了 300 个真空电子管执行算术与逻辑运算，开创了一个计算机的新时代。

1.1.2 电子数字计算机

电子计算机源于 20 世纪电子领域基础研究的突破，目前，以微电子技术为基础制造的电子计算机已经完全取代了机械计算机和机电（继电器）计算机，因此电子计算机也直接简称为计算机。依据信息的表达形式不同，电子计算机可分为电子模拟计算机和电子数字计算机。电子模拟计算机采用连续的模拟电信号表达信息，以运算放大器等模拟电路处理模拟电信号，计算精度取决于模拟器件，而且必须通过手动更改模拟电路才能改变处理过程，性能难以提高，目前只应用在某些专业领域。电子数字计算机采用离散的数字量表达和处理信息，计算精度有保证，易于实现可编程控制，因此成为发展的主流。现在所说的计算机一般都是指电子数字计算机。今天，电子数字计算机的用途也不再仅限于数值计算，强大的计算能力和极高的计算速度使得计算机已经成为文字、图形、声音、影像等多种信息处理的有力工具。

电子数字计算机的发展与电子技术的进步紧密相关，一般划分为 4 个时代。

1. 电子管时代（1945—1955 年）

最初的电子计算机，如阿塔纳索夫-贝瑞计算机、ENIAC（Electronic Numerical Integrator And Computer，电子数字积分计算机）等，以真空电子射线管作为关键器件。电子管的工作机理决定了其成品体积大、质量大、易碎、制造成本高，电子管计算机不仅速度慢，而且体积庞大、耗电量巨大、对环境要求苛刻、可靠性差，制造、运行和维护的费用都很高，因此电子管计算机只有少量研究机构使用，基本用于数值计算，如计算三角函数表等。

2. 晶体管时代（1955—1965 年）

1947 年半导体晶体管发明以后，有了电子管的替代品，由于晶体管比电子管体积小、质量小、速度快、耗电省、造价低，且性能更可靠，所以得到大规模商业化生产，使计算机的制造和维护成本大大降低。晶体管时代，计算机的使用范围在研究机构中进一步扩大并进入一些大型企业，主要用于科学和工程计算，如偏微分方程的求解。

3. 集成电路时代（1965—1970 年）

1958 年出现了将成千上万个晶体管制造在一片半导体晶片上的集成电路（IC，Integrated Circuit），使得计算机在体积、质量、速度、功耗、可靠性等指标上大幅提高，制造成本大大降低。集成电路时代，计算机的应用范围迅速扩大到各个领域，用途不再局限于数学计算，还应用于数据处理、文字处理等多个方面。

4. 大规模集成电路时代（1970—1978 年）、超大规模集成电路时代（1978 年至今）

大规模集成电路（Large-Scale Integrated Circuit）、超大规模集成电路（Very Large-Scale Integrated Circuit）的出现使计算机的性价比进一步提高，并且使计算机的发展形成了两个大的分支。

一个分支是巨型计算机（Super Computer），巨型机也称超级计算机或高性能计算机，并无明确定义，一般是指在当代一定时期内计算能力最强的最高性能计算机。巨型机耗资巨大，代表了一个国家的最高计算机技术，是综合国力的体现。目前，巨型机的处理速度已达浮点运算每秒百万亿次的数量级。巨型机主要应用在天文、天气预报、量子化学、核物理等科学计算领域，人工智能、虚拟现实等研究领域，以及政府统计、企业财务管理等数据处理领域。

另一个分支是微型计算机（Microcomputer），微机是最近 30 多年来依赖 LSI、VLSI 技术发展起来的通用计算机，以其体积小、质量小、性价比高等特点，被很快应用到生产管理、自动控制、经营管理和个人事务等广泛领域。现在的微机的性能已经远远胜过 30 年前的巨型机。

这个时代的一个标志性事件就是集成了控制单元和运算单元的集成电路芯片的诞生，也就是现在频繁提及的两个名词——中央处理器 CPU（Central Processing Unit）和微处理器（Microprocessor）。随着以此为核心的微机的普及，颠覆性地改变了通信、传媒等行业的传统模式，催生出众多新兴行业和高科技产业，大大提高了社会生产率和科技创新能力，彻底改变了整个世界，影响了人类的生活方式。

1.1.3 微处理器

作为微机的核心，微处理器在 30 多年的时间里更新换代的速度超出了大多数人的想象。

通常，有两个重要指标可以衡量微处理器的性能：一个是字长（Word Length），指微处理器一次能计算的二进制数据（称为字，Word）的位数，表征微处理器的运算能力；另一个是时钟频率，指触发微处理器工作的时间节拍，从一定程度上表征微处理器的工作速度。

对于集成电路而言，更高的集成度意味着电子传送距离更短、工作频率更高、功耗更低且制造成本更低。更高的频率意味着更好的性能。Intel 创始人戈登·摩尔（Gordon Moore）在 1965 年曾预测集成度的增长趋势，被称为摩尔定律：每隔一年芯片中的晶体管会翻番，1975 年修订为每隔两年芯片中的晶体管会翻番。之后的 30 多年，这一预测奇迹般地被事实所验证。表 1.1 所示为最有影响力的微处理器制造商 Intel 的几款不同时期产品的对比，可具有代表性地说明微处理器不断更新的发展历程。

集成电路的集成度依赖于制造工艺，制造工艺一般以逻辑门电路线宽来描述，目前主流微处理器采用的是 45 nm 工艺。从理论角度讲，硅晶体管集成工艺还能够继续缩小到 4 nm 的级别，这时晶体管将因为电子漂移（Electrons Drift）而无法控制电子的进出。如果到那时还找不到替代半导体晶体管的材料，摩尔定律便会失效。一些可能的新材料，碳纳米管（Carbon Nanotubes）、硅纳米线（Silicon

Nanowires）、分子开关（Molecular Crossbars）、相态变化材料（Phase Change Materials）、自旋电子（Spintronics）等目前都处于实验阶段，今后计算机的发展还要拭目以待。

表 1.1 Intel 微处理器产品

年　份	1971	1974	1978	1985	2007
典型微处理器	4004	8080	8086	80386	QX9650
字长（位）	4	8	16	32	64
时钟频率	108 kHz	2 MHz	4.77 MHz	16 MHz	3 GHz
线宽	10 μm	6 μm	3 μm	2 μm	45 nm
芯片集成度（晶体管/片）	2.3×10^3	6×10^3	29×10^3	275×10^3	410×10^6

人类对未知世界的探索永无止境，对新一代计算机的研究也从未停止过，基于新技术、新理论的另一个计算机时代必将到来，通过生命物质和量子特性来实现计算机技术的发展，也许是目前最具想象力的解决方案。

1.2 计算机中的信息编码

1.2.1 二进制编码

计算机是处理信息的机器，除了数值信息外，还有文字、符号、声音、图像等不同种类的信息，那么在计算机中如何表达这些信息呢？

数字计算机使用数字信号表达信息。若干位数字的组合可以表达多种信息而又不混淆，称为编码。例如，5 位十进制数的组合有 $10^5=100\,000$ 种，即可以表达 100 000 种信息而不混淆。这样按约定形成的表达信息的编码称为数据，显然，数据是信息在计算机中的表达形式。

同理，数字计算机的指令（控制计算机执行某种操作的命令）也是由数字编码表达的。从广义上讲，指令编码也是数据。

自从楚泽在 Z-3 计算机中首先采用二进制计数以来，现代电子数字计算机都采用二进制编码。之所以采用二进制，而不采用人们常用的十进制或其他进制编码，有以下两个原因。

（1）二进制是最基本的进位计数系统，容易表达。二进制只有两个基本数字：0 和 1，是基本数字最少的进制。计算机内的电路只要能实现两种状态的表达（如三极管的饱和导通和截止、继电器的通和断、磁片的磁化和消磁、光盘对光的反射和不反射等），都可以实现二进制数的表达。而我们习惯的十进制有 10 个基本数字，0～9，在计算机中实现 10 种状态的表达显然困难得多。

（2）二进制运算规则简单，可以通过逻辑和移位电路实现。二进制数加法通过逻辑异或实现，进位可以通过逻辑与实现；二进制数减法可以通过转化为加补码实现；二进制数乘法可以通过部分积右移加被乘数实现；二进制数除法可以通过部分余数左移减除数实现。而要实现十进制数的运算规则，实现电路就复杂得多了。

因此，现代电子数字计算机中存储和处理的所有信息都采用二进制编码。例如，自然数 5 的 8 位无符号数编码是 00000101B（后缀 B 表示这是二进制数字），整数–9 的 8 位符号数编码是 11110111B，汉字“杨”的 Unicode 编码是 0110011101101000B。在计算机中表达不同类型的信息时，采用不同的编码形式，后面将介绍几种常用的编码。针对数值信息（数学值）的编码，为了区分数值本身及其编码，一般将表达数值的编码称为机器数，将数值本身称为真值。

我们会发现，虽然二进制编码表达简单，但其内容和位数却不方便识别，由于 4 位二进制数相当

于一位十六进制数（$2^4=16$），所以在本书后续章节中，会经常采用十六进制数代替二进制编码。例如，上面的三个二进制编码分别为 05H、0F7H 和 6768H，其中后缀 H 表示这是十六进制数字。要注意的是，最高位为 A～F 的十六进制数前面要添加一个“0”，这是为了与其他符号区分开，其他符号规定不以 0～9 开头。这样，可以很快判断出上面三个十六进制数分别是 8 位、8 位和 16 位二进制编码，如果需要，可以很快转化出相应的二进制编码。必须注意，计算机中并不存在这些十六进制数，写成十六进制数只是为了方便识别二进制编码的内容和位数。

1.2.2　整数的编码

计算机中定义两类整数：无符号数和符号数。

1．无符号数编码

无符号数没有符号，不用考虑符号的编码，所以实际上无符号数编码直接使用真值的二进制形式。8位无符号数编码表如表1.2 所示，可以表达 0～255 共 256（8 位编码，$2^8=256$）个无符号数。

表 1.2　8 位无符号数编码表

无符号数	无符号数编码（8 bit）
0	00000000
1	00000001
2	00000010
…	…
127	01111111
128	10000000
…	…
255	11111111

由于无符号数编码就是真值，显然，无符号数编码可以直接进行算术运算。如：

$$[64]_{无}+[100]_{无}=01000000B+01100100B$$
$$=10100100B=[164]_{无}$$

计算机的字长是有限的，有限位数的无符号数编码也只能表达一定范围的无符号数，如果运算结果超出这个范围，就会产生错误，这种情况称为溢出。例如，以下计算是错误的：

$$[200]_{无}+[100]_{无}=11001000B+01100100B$$
$$=00101100B=[44]_{无}\text{（错误）}$$

由于 8 位无符号数编码表达的真值范围是 0～255。300 超出了无符号数所能表达的范围，没有对应的 8 位无符号数编码，上面这个加法得到的结果肯定是错误的。无符号数编码的加法/减法是否溢出，可以通过最高位的进位/借位来判别。

2．符号数编码

符号数编码不仅要表示真值绝对值，而且要表示符号。曾经使用和正在使用的有 4 种编码方法：原码（Signed Magnitude），反码（One's Complement），补码（Two's Complement），以及移码（Excess-N or Biased Representation）。

表 1.3　8 位原码编码表

符号数	原码
–127	11111111
…	…
–3	10000011
–2	10000010
–1	10000001
–0	10000000
+0	00000000
+1	00000001
+2	00000010
+3	00000011
…	…
+127	01111111

（1）原码

X 为真值，n 位原码的定义为：

$$[X]_{原}=|X| \qquad 0\leqslant X<2^{n-1}$$
$$[X]_{原}=2^{n-1}+|X| \qquad -2^{n-1}<X\leqslant 0$$

从原码定义可以看出：原码的最高位表示符号（非负数表示为 0，非正数表示为 1），其他位是真值绝对值。8 位原码编码表如表 1.3 所示，可表达–127～+127 共 255 个符号数。

一些早期计算机（例如 IBM 7090）使用原码表示法。其特点如下。

① 乘、除运算比较方便，可以独立地对符号和真值绝对值分别处理。

② 加、减运算比较复杂，首先需通过两个原码的符号，确定实际要做加法还是减法，然后可能还需要比较真值绝对值的大小，才能判别出结果的符号。

$[+5]_{原}+[-9]_{原}=?$

$[+5]_{原}=00000101B$　$[-9]_{原}=10001001B$

两个原码的符号位不同，实际做减法。真值绝对值0000101B＜0001001B，结果符号为负。所以

$$[+5]_{原}+[-9]_{原}=2^{n-1}+0001001B-0000101B=10000100B=[-4]_{原}$$

③ 0的原码有两个，判断是否为0要比较两次。

$[+0]_{原}=00000000B$　$[-0]_{原}=10000000B$

（2）反码

由于加、减运算是算术逻辑单元的基本操作，操作频率远高于乘、除运算，而原码的符号位不参与运算，需要单独处理。为了提高数据处理效率，解决符号位参与加、减运算的问题，引入了反码。

X为真值，n位反码的定义为：

$$[X]_{反}=|X| \qquad 0\leqslant X<2^{n-1}$$

$$[X]_{反}=2^{n}-1-|X| \qquad -2^{n-1}<X\leqslant 0$$

从反码定义可以看出：非正数的反码是其真值绝对值的1补数；也可以在真值绝对值的基础上按位求反得到，故得名反码。8位反码编码表如表1.4所示，可表达–127～+127共255个符号数。

表1.4　8位反码编码表

符号数	反码
–127	10000000
…	…
–3	11111100
–2	11111101
–1	11111110
–0	11111111
+0	00000000
+1	00000001
+2	00000010
+3	00000011
…	…
+127	01111111

一些老式计算机使用反码表示法，如PDP-1，CDC 160A，UNIVAC 1100/2200系列等。其特点如下。

① 符号位一起参与加、减运算。但反码加法，进位需要送回到最低位再加（循环进位）；反码减法，借位需要送回到最低位再减（循环借位）。

$[X_1]_{反}+[X_2]_{反}=[X_1+X_2]_{反}$

$[9]_{反}+[5]_{反}=00001001B+00000101B+$ 进位 $=00001110B+0=00001110B=[14]_{反}$

$[X_1]_{反}-[X_2]_{反}=[X_1-X_2]_{反}$

$[5]_{反}-[9]_{反}=00000101B-00001001B-$ 借位 $=11111100B-1=11111011B=[-4]_{反}$

② 减法运算可转换为加法，简化了算术逻辑单元的设计。

因为$[-X]_{反}+[X]_{反}=2^{n}-1-|X|+|X|=2^{n}-1$，故$[-X]_{反}=2^{n}-1-[X]_{反}=([X]_{反})_{反}$

$[X_1-X_2]_{反}=[X_1]_{反}-[X_2]_{反}=[X_1]_{反}+[-X_2]_{反}=[X_1]_{反}+([X_2]_{反})_{反}$

$[9]_{反}-[5]_{反}=[9]_{反}+([5]_{反})_{反}=00001001B+(00000101B)_{反}+$ 进位

$=00001001B+11111010B+$ 进位

$=00000011B+1=00000100B=[4]_{反}$

③ 0的反码仍然有两个。

$[+0]_{反}=00000000B$　$[-0]_{反}=11111111B$

（3）补码

反码较好地解决了符号参与运算的问题，但由于循环进位需要两次算术相加，延长了计算时间，为进一步简化算术逻辑单元的设计，引入了补码。

X为真值，n位补码的定义为：

$[X]_{补}=|X|$　　$0 \leqslant X<2^{n-1}$

$[X]_{补}=2^n-|X|$　　$-2^{n-1} \leqslant X<0$

表 1.5　8 位补码编码表

符号数	补码
–128	10000000
–127	10000001
…	…
–3	11111101
–2	11111110
–1	11111111
0	00000000
+1	00000001
+2	00000010
+3	00000011
…	…
+127	01111111

从补码定义可以看出：负数的补码是其真值绝对值的 2 补数；也可以在真值绝对值的基础上按位求反加 1 得到。8 位补码编码表如表 1.5 所示，可表达–128～+127 共 256 个符号数。

当代计算机都使用补码表示法。其特点如下。

① 同反码一样，补码的符号位一起参与加、减运算，并且回避了反码的循环进位和循环借位。

$[X_1]_{补}+[X_2]_{补}=[X_1+X_2]_{补}$

$[9]_{补}+[5]_{补}=00001001B+00000101B=00001110B=[14]_{补}$

$[X_1]_{补}-[X_2]_{补}=[X_1-X_2]_{补}$

$[5]_{补}-[9]_{补}=00000101B-00001001B=11111100B=[-4]_{补}$

② 同反码一样，补码的减法运算可转换为加法。

因为$[-X]_{补}+[X]_{补}=2^n-|X|+|X|=2^n$，故$[-X]_{补}=2^n-[X]_{补}=([X]_{补})_{反}+1$

$[X_1-X_2]_{补}=[X_1]_{补}-[X_2]_{补}=[X_1]_{补}+[-X_2]_{补}=[X_1]_{补}+([X_2]_{补})_{反}+1$

这里要注意：$[-128]_{补}$其实没有相应的 2 补数$[128]_{补}$（因为$-2^{n-1} \leqslant X< 2^{n-1}$），但只要加法运算不溢出，结果也是正确的。

$[9]_{补}-[5]_{补}=[9]_{补}+([5]_{补})_{反}+1=00001001B+(00000101B)_{反}+1$

$=00001001B+11111010B+1$

$=00000100B=[4]_{补}$

$[9]_{补}-[-5]_{补}=[9]_{补}+([-5]_{补})_{反}+1=00001001B+(11111011B)_{反}+1$

$=00001001B+00000100B+1$

$=00001110B=[14]_{补}$

$[-9]_{补}-[-128]_{补}=[-9]_{补}+([-128]_{补})_{反}+1=11110111B+(10000000B)_{反}+1$

$=11110111B+01111111B+1$

$=01110111B=[119]_{补}$

③ 0 的补码只有一种表示。

$[0]_{补}=00000000B$

④ 补码同样也存在溢出问题，但溢出的判断很简单，如：

$[+120]_{补}+[+16]_{补}=01111000B+00010000B=10001000B=[-120]_{补}$（错误）

当补码加法/减法操作后，如果最高位进位/借位与次高位进位/借位不相同，表示运算结果溢出，否则表示无溢出，这种判别方法也称为双高异或判别法。比如 120 + 16 时，次高位有进位为 1，最高位没有产生进位为 0，1 异或 0 = 1，说明有溢出。

（4）移码

X 为真值，n 位移码的定义为：

$[X]_{移}=2^{n-1}+[X]_{补}$　　$-2^{n-1} \leqslant X<2^{n-1}$

从移码定义可以看出：移码是在补码基础上加一个偏移量（Bias）得到的。

如果是 8 位移码，则：

$[+9]_{移}=10001001B$　$[+5]_{移}=10000101B$

$[-9]_{移}=01110111B$　$[-5]_{移}=01111011B$

引入移码的目的是用于浮点数的指数编码，这样指数比较时不必比较符号位，并保证0的编码为全0（参见具体浮点编码标准）。

1.2.3 实数的编码

实数中包含小数点，可表示为若干等价形式。作为规格化（Normalization）形式，任何一个不为0（0一般都作为特例编码）的实数X都可唯一表示为：

$$X=\pm M\times R^{E} \qquad R^{-1}<|M|<1$$

M为小数形式，称为尾数（Mantissa or Significand），代表X的全部有效数字；

R为采用的进位制的基（Base or Radix）；

E为整数形式，称为指数（Exponent），代表X的小数点实际位置。

如一个实数–13.6640625用二进制可唯一表示为：

$$-13.6640625=-0.136640625\times10^{2}=-1101.1010101\text{B}=-0.11011010101\text{B}\times2^{4}$$

计算机中的实数编码有两种形式：定点（Fixed Point）和浮点（Floating Point），区别在于是否固定指数，即小数点的位置。

1．定点编码

定点编码定义指数E不变，即小数点在尾数中的位置固定（一般定义在尾数数值部分的最前面或最后面），因此只需要编码符号和尾数。定点编码表达的实数称为定点数，采用定点编码的计算机称为定点机。

假定8位定点编码的最高位是符号位，低7位为尾数数值部分的原码，定义指数$E=0$，即小数点固定在尾数数值部分的最前面。则一个二进制实数–0.1011001B的定点编码为11011001B。

定点机电路实现简单，但由于指数固定，表达的定点数范围非常小。如上面的编码，除0以外，表达范围为 |0.0000001B|～|0.1111111B|，表达范围之外的实数，在计算前需要用比例因子将其转化为规定的定点数（规格化），计算后再用比例因子将结果转化为实际值，为了保持计算精度和不溢出，计算过程中可能需要多次调整比例因子。

2．浮点编码

浮点编码除符号和尾数外，还要编码指数（称为阶码），用于指示小数点在尾数中的位置，即小数点可“浮动”。浮点编码表达的实数称为浮点数，采用浮点编码的计算机称为浮点机。

我们假定8位浮点编码的最高位是符号位，中间3位为指数的补码，低4位为尾数的原码。则一个二进制实数$-0.01011001\text{B}=-0.1011001\text{B}\times2^{-1}$的浮点编码为11111011B。这种编码除0以外，表达范围为 $|0.1000\text{B}|\times2^{-4}$～$|0.1111\text{B}|\times2^{3}$，相对定点编码，在相同存储空间，浮点编码表示的数值范围大（指数可变化），但付出了一些精度代价（尾数数值部分编码位数减少）。浮点计算机构造复杂，但表达范围大，运算精度高。

早期微机以定点处理器为主，随后又出现了“定点处理器+浮点协处理器”的形式，现在已经普遍采用浮点处理器。浮点编码格式曾经出现过多种，目前都基本统一为IEEE754标准（IEEE754指数编码采用移码，移码的定义和尾数规格化与本书介绍不同，具体编码格式请参见相关资料）。

1.2.4 十进制数的编码

十进制数的二进制编码称为BCD（Binary Coded Decimal）码。十进制有10个基本数字，4位二进制编码有$2^{4}=16$种组合，原则上可任意定义其中的10种组合来表示十进制的10个基本数字，但通常选择前10个组合。根据需要，也可以用8位二进制数来编码十进制数。这两种编码分别称为压缩型

十进制的二进制编码（Packet BCD）和非压缩型十进制的二进制编码（Unpacket BCD），如表 1.6 所示。BCD 可以进行算术运算，但结果需要调整（具体调整过程可参见后续章节）。

表 1.6　BCD 编码表

十进制数	Packet BCD	Unpacket BCD
0	0000	00000000
1	0001	00000001
2	0010	00000010
3	0011	00000011
4	0100	00000100
5	0101	00000101
6	0110	00000110
7	0111	00000111
8	1000	00001000
9	1001	00001001

1.2.5　英文字符的编码

美国国家标准学会（ANSI）1963 发布了《美国信息交换标准代码》（ASCII，American Standard Code Information Interchange），主要用于统一当时不同计算机厂商使用的字符集，现在 ASCII/ISO646/GB1988 也仍然是最通用的编码系统。受当时设备所限，ASCII 码采用 7 位编码，表达 128 个字符，其中包括英文大写字母 26 个、英文小写字母 26 个、十进制数字 10 个、常用符号 33 个和通用控制符号 33 个（参见附录 A）。

1.2.6　汉字的编码

“信息交换汉字编码字符集基本集”（GB2312—1980）是我国发布的第一个汉字编码标准。GB2312 采用 2 字节（每字节 7 位）编码，共计字符 7445 个，其中包括简体汉字 6763 个、一般符号 202 个、序号 60 个、数字 22 个、拉丁字母 52 个、日文假名 169 个、希腊字母 48 个、俄文字母 66 个、汉语拼音字母 37 个。GB2312—1980 是几乎所有的中文系统和国际化的软件都支持的中文字符集，也是最基本的中文字符集。

1.2.7　多文种的编码

为便于多个文种的同时处理，对世界上的所有文字统一编码，国际标准化组织和 Unicode 协会共同发布了新的编码字符集标准 ISO/IEC10646.1: 1993《信息技术通用多 8 位编码字符集（UCS，Universal Multiple-Octet Coded Character Set）第一部分：体系结构与基本多文种平面》，相应的国家标准是 GB13000.1—1993。它采用 4 字节（每字节 8 位）编码，这 4 字节分别表示组、平面、行和字位（128 组×256 平面×256 行×256 字位）。目前实现的是 00 组的 00 平面，称为“基本多文种平面”（BMP，Basic Multilingual Plane），编码位置 65536 个。由于基本多文种平面所有字符编码的前两个字节都是 0，目前在默认情况下按照 2 字节处理。

1.3　计算机运行原理

1.3.1　计算机的定义

1．自动计算的机器

什么是计算机（Computer）？算盘是计算机吗？对数计算尺、能进行四则运算的计算器是计算机吗？对于计算机，从不同角度有不同的描述，很难有一个公认的严密定义。我们可以简单认为：计算机是可以自动计算（即自动处理数据）的机器。从这个角度来说，算盘和对数计算尺不是计算机，因为如果没有人脑参与，它们就无法完成计算过程，也得不出结果。而计算器在输入算式后按“=”键，