职工高等工业专科学校教材

微型计算机原理及应用

MCS-51系列单片机

第3版

王义方
周伟航 主编

机械工业出版社

职工高等工业专科学校教材

微型计算机原理及应用
——MCS-51 系列单片机

第 3 版

王义方
周伟航 主编

机 械 工 业 出 版 社

本书系统而详尽地介绍了MCS-51系列单片微型计算机的软件、硬件及其应用。内容包括：微机工作原理、基本构成及单片机的结构特点；MCS-51系列单片机指令系统及汇编语言程序设计；并行I/O口及系统扩展方法；定时器/计数器及其应用；中断系统；串行口原理和实用通信程序；应用举例等。

本书注重理论和实践的结合，书中配有丰富的例题、习题与思考题，书末附有与本书配套的实验指导。内容简明扼要，通俗易懂，便于教学和自学。

本书适用于职工大学和业余大学，大、中专和各类微机培训班也可选用，并可供广大工程技术人员自学入门之用，尤其适合于成人教育。

图书在版编目（CIP）数据

微型计算机原理及应用：MCS－51系列单片机/王义方，周伟航主编．－3版．－北京：机械工业出版社，1997.11

职工高等工业专科学校教材

ISBN 7-111-05504-7

Ⅰ．微…　Ⅱ．①王…②周…　Ⅲ．单片微型计算机－高等学校：专业学校-教材　Ⅳ．TP368.1

中国版本图书馆CIP数据核字（97）第03136号

机械工业出版社（北京市百万庄大街22号　邮政编码100037）

责任编辑：贡克勤　版式设计：张世琴　责任校对：罗莉华

封面设计：姚　毅　责任印制：付方敏

北京市密云县印刷厂印刷　·　新华书店北京发行所发行

2002年6月第3版第18次印刷

787mm×1092mm 1/16 · 14印张 · 334千字

215 546—218 545册

定价18.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

本社购书热线电话(010)68993821、68326677-2527

封面无防伪标均为盗版

第1版前言

本书是职工高等工业专科学校工业电气自动化专业的试用教材，是根据教育部成人教育司、全国总工会和机械工业部教育局制订的“工业电气自动化专业”教学计划和“微型计算机原理及应用”教学大纲而组织编写的。

本书适用于职工大学和业余大学。中等专业学校及各种微机培训班也可以选用，并可供广大工程技术人员自学入门之用。书中带星号（*）的章节，可根据不同需要而灵活掌握，决定取舍。

本书主要介绍目前在我国使用最广泛的Z80微处理器的原理及Z80单板机的应用。全书着眼于应用，力求简明扼要，通俗易懂。每章均附有较多数量的例题、思考题和习题，书末还附有应用实例，便于教学与自学，尤其适合于成人教学。

本书由苏州市职工业余大学王义方同志（第一、二、三、四、五、六章）及上海市卢湾区业余大学杨振英同志（第七、八、九章）编写，全书由王义方同志主编，由湖南大学杨润生副教授主审。

本书编写后经过试用，于1984年8月由湖南大学、交通大学、汽车制造工业学院和核工业部九院职工大学等十六所高校所组成的审稿会审定通过。参加审稿的有田端庭、莫松涛、白英彩、王志清、吴誉、邱振诒、姚国定、王旭、庄婉娜、李炳延、王华兴和温良玉等同志。

本书在编写过程中曾得到劳定坳、张崑藏、冯子纲、张柏雄、张端宁和丁建强等同志的热情帮助和支持，在此一并表示衷心的感谢。

由于编者水平有限，编写时间仓促，书中一定存在不少缺点和错误，恳请读者批评指正。

编　者

1984年10月

第 2 版前言

本书是原《微型计算机原理及应用》一书的修订本。原书是 1984 年根据教育部成人教育司、全国总工会和机械工业部教育局制订的“工业电气自动化专业”教学计划和“微型计算机原理及应用”教学大纲而组织编写的，是职工高等工业专科学校工业电气自动化专业的正式教材。自 1985 年出版发行以来，得到了广大读者的关心和支持，许多大、中专学校和各种类型的微机培训班都选它作为教材，社会反映良好，因而年年印刷出版，在此，作者深表感谢。

根据许多读者来信提出的宝贵建议，现在对该书进行了修订，以进一步适应形势发展的需要。

这次修订主要增加了 MCS-51 单片微机（安排在第九章）的有关内容，另外对原各章内容都作了必要的修订，第二章删去了“其他微处理器简介”一节，第三章删去了部分例子，第五章全部进行了重写，简化了有关存储器芯片的内部结构的内容，增写了许多新的常用芯片的内容，第六章增写了中断的分类，第七章删去了“线路反转”技术识别按键和利用 D/A 转换芯片实现 A/D 转换的内容，第八章删去了“EPROM 写入接口电路与程序”及“盒式磁带机串行接口电路与程序”等两节，增写了“Z80 单板机的扩展”一节，原第九章改为第十章并删去了“保温瓶的自动检测”一节，增写了“应用系统设计方法简述”及“定长控制系统”等内容，最后还对原书中的疏漏和错误进行了修正。

本书修订后，不但继续保持了简明扼要、通俗易懂和便于教学与自学的特点，而且增加了许多新的芯片及一些设计技术的内容，在层次上有所提高，因而更加精炼、实用和先进。按本专业教学计划的规定，Z80 微处理器的原理及应用部分讲课为 72 学时，实验为 14 学时，MCS-51 单片微机的原理及应用部分则建议为 60 学时（其中带星号‘*’的章节为选讲内容），实验为 10 学时，各校根据具体情况可酌情增减。

本书由苏州市职工业余大学王义方任主编，并编写第一、二、三、四、五、六章及第十章的§10-1、§10-2，上海市卢湾区业余大学杨振英编写第七章、第八章的§8-1、§8-2、§8-3 及第十章的§10-3、§10-4、§10-5，苏州市职工业余大学丁建强编写第八章的§8-4 及第十章的§10-6，苏州市职工业余大学李绍成编写第九章。

本书修订后，于 1991 年 8 月由九所高校和厂职工大学所组成的审稿会审定通过。由湖北汽车工业学院田瑞庭教授主审，桂林市职工业余大学邱振诒副教授协审。参加审稿的还有成都市职工大学聂剑和徐文芳、天津市红桥区职工大学刘风章、杭州市工人业余大学王华兴、淄博市职工大学吕昌泰、常熟市高等专科学校周伟航、德阳市第二重机厂职工大学温良玉和佛山市电梯厂李思亲等。

本书在修订过程中曾得到苏州市职工业余大学有关领导的支持，彭玲再次描画了书中的插图，在此一并表示衷心的感谢。

由于编者水平有限，书中难免存在缺点和错误，恳请读者批评指正。

编　者

1992 年 10 月

第3版前言

本书是原《微型计算机原理及应用》一书（1992年10月第2版）的修订版。

如原书第2版前言所述：该书自出版发行以来，得到了广大读者的关心和支持，许多大、中专学校和各种类型的培训班都选它作为教材，社会反映良好，因而年年印刷出版。

随着我国科学教育事业的蓬勃发展，考虑到以Z80-CPU为核心的单板计算机已经落伍，而MCS-51单片机已在我国成为主流系列，并且正在不断发展，为使本书跟上时代步伐，我们这次对原书作了全面的修订和改写，以满足广大读者的要求，并感谢大家对本书的厚爱。

这次修订，主要删去了原书中有关Z80内容的第二章、第三章、第五章、第六章、第七章、第八章及第十章的部分内容；保留了原书第四章中的大部分程序设计应用举例并增加了反映MCS-51单片机特点的若干例子；同时增写了实验指导一章。

全书用MCS-51系列单片机作为典型机，由浅入深、循序渐进地介绍了单片微型计算机系统的结构、工作原理及指令系统，介绍了汇编语言程序设计的基本理论、方法及其应用，对MCS-51系列单片机的系统扩展、定时/计数电路、中断系统、串行口的原理和应用等，也作了较详尽的介绍。

鉴于本版以MCS-51系列单片计算机为主要内容，故这次在原书名上增加了一个副标题——“MCS-51系列单片机”。在编写过程中，尽量与原书的风格一致，力求保持原书简明扼要、通俗易懂和便于教学与自学的特点；书中配有丰富的例题，每章末附有习题与思考题；书末的实验指导部分提供了与本书内容相配套的9个基本实验，这些都有助于读者从理论和实践的结合上，更好地理解和掌握MCS-51系列单片机的原理及其应用。

按照“工业电气自动化专业”教学计划的要求，本书讲课学时数建议为68学时左右，实验为14学时，各校可根据情况酌情增减。

本书由苏州市职工业余大学王义方和常熟高等专科学校周伟航任主编。由王义方编写第一章、第十章的第一节、第二节，周伟航编写第二章、第三章、第四章、第五章（根据原书第四章思路和格式编写）及第九章，常熟高等专科学校陈志刚编写第六章，顾启明编写第七章、第八章，苏州市职工业余大学丁建强编写第十章的第三节，实验指导部分由周伟航、顾启明、陈志刚合作完成。

本书由苏州市广播电视大学、苏州市职工大学冯子纲副教授主审，上海交通大学顾云云副教授、陆景明副教授协审。他们对本书原稿的审阅付出了艰辛的劳动，并提出大量宝贵意见。在本书编写过程中，得到了常熟高等专科学校、苏州市职工业余大学有关领导和丁建强、李绍成等同志的大力支持，并听取了包括读者在内各方面有关人士的意见，在此一并表示衷心的感谢。

由于编者水平有限，书中难免存在缺点和错误，恳请读者批评和指正。

编　者

1996年10月

目　录

第一章　计算机的基础知识

第一节　概　　述

本章主要介绍微处理器及微型计算机（简称微机）的发展与应用、计算机的数制、码制和微型机的组成及其工作原理。要求能进行十进制、二进制和十六进制数之间的相互转换，明确补码在微机中的应用，了解微机的基本结构和工作过程。

一、微型计算机的发展概况

随着计算机技术和大规模集成电路的发展，微型计算机应运而生。自从美国 Intel 公司于 1971 年推出以 I4004 微处理机为核心的 4 位微机以来，在短短的十几年时间里获得了飞速发展：微处理器（CPU）的集成度差不多每两年翻一番，且其功能增长一个数量级。纵观微机发展历史，至今已经经历了四代的演变：

第一代（1971 年～1972 年）

美国 Intel 公司首先推出 I4004，它是 4 位的微处理器。以它为核心再配上相应的 RAM、ROM 和输入、输出（I/O）接口电路就构成了 MCS-4 微计算机。同年，该公司还研制出 8 位的微处理器 I8080。

第二代（1973 年～1975 年）

第二代微机的代表产品是美国 Intel 公司的 I8080、MCS-80 和 Motorola 公司的 M6800，它们均为 8 位的中档微机。

1976 年至 1977 年，美国 Zilog 公司研制的 Z80 和 Intel 公司研制的 I8085，是高性能的 8 位微处理器，一般称之为第二代半的产品。

第三代（1978 年～1980 年）

代表产品是美国 Intel 公司的 I8086、Zilog 公司的 Z8000 和 Motorola 公司的 M68000，它们是 16 位微处理器，又称第一代超大规模集成电路的微处理器。

第四代（从 1981 年开始）

代表产品是美国 Intel 公司的 Iapx432、Bell 研究所的 MAC-32、NS 公司的 NS16032，它们是 32 位微处理器，又称超级微处理器。

二、微型计算机的特点

微型计算机具有普通计算机所具备的特点，如：

1. 运算速度快

由于计算机是一种能存储程序且能执行程序的自动、高速、精确地进行工作的电子装置，其运算速度现在已经达到微秒级甚至更高，使得许多过去无法快速处理的复杂问题得以及时处理。

2. 运算精度高

电子计算机具有过去计算工具无法比拟的运算精度，一般可达到十几位甚至几十位。

3. 具有“记忆”和逻辑判断能力

计算机中的存储器具有存储和“记忆”大量信息的能力，能存储输入的程序和数据，保存计算和处理的结果。同时，计算机还能进行逻辑判断，并根据判断结果自动地确定下一步应该做什么，从而使计算机能解决各种不同的问题，具有很强的通用性。

4. 可靠性高

微电子技术和计算机科学技术的发展，使现代计算机和微机连续无故障运行时间可达几万、几十万小时以上，即可连续几个月甚至几年工作而不出差错，从而具有很高的可靠性。

5. 价格廉、体积小、质量轻、功耗低

这一特点使得一些中小型的或廉价的设备都能配备微处理器或微型计算机，从而使它深入到过去计算机所无法深入的领域。

6. 方便灵活，通用性强

微型计算机采用总线结构形式，能根据需要来选择总体布局，十分机动灵活地构成各种各样的系统：小到将微处理器作为一个部件组装在应用设备中，使设备电脑化；大到与大型计算机和外围设备连成一个网络，彼此进行通信并资源共享。由于目前构成微机的基本部件大多已经标准化，又能十分方便地进行扩展与集散，所以适应面广，可以面向各行各业。

三、微型计算机的应用

计算机的应用已经渗透到人类社会生活的各个领域，并将与每个人的生活发生越来越密切的联系。其应用归纳起来主要有以下几个方面：

1. 数值计算

数值计算又称科学计算，是指将微机用于完成科学研究和工程设计及日常生活中所提出的数学问题的计算。

2. 信息处理

当今社会已经从工业社会进入信息社会。所谓信息是指人们从事社会生产和生活中所形成的数字、文字、语言、符号、图象等的总称。人们必须及时收集、分析、加工、处理大量信息，这是信息社会的特征之一。微机是进行数据处理、企业管理、情报检索及办公自动化等方面的信息处理工作的最得心应手的工具。

3. 实时控制

实时控制又称过程控制，是指用计算机实时地采集检测数据，按最佳方案对控制对象进行自动控制或自动调节。利用微机进行过程控制已在冶金、石油、化工、纺织、机械、军事等许多部门得到广泛的应用，它不仅大大提高了控制的及时性和准确性，而且能极大地改善劳动条件、提高产品质量、节约能源、降低成本，是微机应用中的一个重要领域。

4. 辅助设计

计算机辅助设计（CAD）是利用计算机的计算及逻辑判断功能，帮助人们进行工程设计和产品设计。计算机辅助设计能使设计过程逐步趋向自动化，大大缩短设计周期，并能节约人力、物力，降低成本，提高设计质量。

5. 人工智能

人工智能是利用微机模拟人类的智能活动：判断、理解、学习、图象识别、问题求解等。例如模拟高水平医学专家进行疾病诊疗的专家系统，以及具有一定“思维能力”的智能机器人等。

还有许多其它方面的应用。总之，现代科学技术的发展，几乎使微机应用进入了一切领域。

第二节　数制与码制

本节主要介绍计算机中数值运算的基本知识，包括进位计数制、不同进位计数制之间的转换、数字与字符的编码方法以及数的符号表示方法等。

一、数制及数制转换

人们通常使用的是十进制数。十进制数的特点是有十个不同的数字：0、1、2、3、…、9，并且是依照“逢十进一”的规则进行运算。

在十进制数中，各位数字所表示的值不仅与该数字本身有关，而且与该数字所在的位置有关。例如数 34，十位上的 3 表示 3 个 10，即 30；个位上的 4 表示 4 个 1，即 4。这里每个位数被赋以一定的位值，称为该位的“权”。十进制数中个、十、百、千…各位的“权”依次为 1、10、100、1000、…，即 10^0、10^1、10^2、10^3、…。每个数位所表示的数字的值应为这个数字与它所对应的权的乘积。

因此，任何一个十进制数均可表示为按权展开的多项式。

例如：$A_nA_{n-1}\cdots A_1A_0.A_{-1}A_{-2}\cdots A_{-m}$

$$=A_n10^n+A_{n-1}10^{n-1}+\cdots+A_110^1+A_010^0+A_{-1}10^{-1}+\cdots+A_{-m}10^{-m}$$

$$=\sum_{i=-m}^{n}(A_i10^i)$$

其中，A_i 是 0～9 十个数字中的任意一个，m、n 为正整数。

这里的 10 被称为基数，它是相邻两数位的权之比。各位的权是基数 10 的整次幂。有时在数的右下方注上基数 10，以便更明确地指出它是一个十进制数。例如 $(34)_{10}$，$(7.1)_{10}$等。

这种“逢十进一”的十进制数是我们平时接触最多、使用最频繁的一种进位计数制。

在计算机中，除了十进制数以外，经常使用的数制是二进制数和十六进制数，在运算中它们遵循的是“逢二进一”和“逢十六进一”的法则，其余与十进制数十分雷同。

（一）二进制数

1. 数的二进制表示及其特点

在二进制数中只有 0 和 1 两个不同的数字，并且是“逢二进一”。它的基数是 2，各位的权是基数 2 的整次幂。与十进制相仿，一个二进制数可表示为按权展开的多项式。

例如：$(10.11)_2=1\times2^1+0\times2^0+1\times2^{-1}+1\times2^{-2}=2.75$

通常在数的右下方注上基数 2，或加后缀 B 表明它是一个二进制数。

例如：$(10111001)_2$ 可写成 10111001B。但对于十进制数，在不会引起误会的场合，可以不加注基数或后缀（十进制数的后缀符号为 D）。

在计算机中的信息均采用二进制数表示，这是因为二进制数具有以下主要特点：

① 二进制数中只有两个数字 0 和 1，因此可用大量具有两个不同的稳定物理状态的元件来表示。例如，可用指示灯的不亮与亮，继电器的断开与接通，磁性元件的反向与正向剩磁，以及脉冲电位的低与高等等，来分别表示二进制数字 0 和 1，这比十进制数的每一位要用具有十个不同的稳定状态的元件来表示，实现起来要容易得多。

② 二进制数的运算简单，大大简化了计算机中运算部件的结构。二进制数的加法和乘法规则，仅为以下八条：

$0+0=0$　　$0+1=1$　　$1+0=1$　　$1+1=10$

$0\times0=0$　　$0\times1=0$　　$1\times0=0$　　$1\times1=1$

③ 二进制数的两个数字 0 和 1 与逻辑代数的逻辑变量取值一致，从而可以采用二进制数进行逻辑运算。这样可以应用逻辑代数作为工具来分析和设计计算机中的逻辑电路，使得逻辑代数成为计算机设计的数学基础。

2. 二进制数转换成十进制数（乘权求和）

例如：把 $(1001.01)_2$ 转换成十进制数。

解　$(1001.01)_2$

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

$=8+1+0.25$

$=9.25$

3. 十进制整数转换成二进制数（除 2 取余法）

如果一个十进制整数 x 已被表示成一个二进制数 $(a_na_{n-1}\cdots a_0)_2$，那么 x 可按二进制的权展开如下：

$x=a_n2^n+a_{n-1}2^{n-1}+\cdots+a_12^1+a_02^0$

不难理解，展开式前 n 项均为 2 的整数倍，而 a_0 的取值非 0 即 1。因此 a_0 即为 x 除于 2 所得之余数。也就是说，x 除于 2 所得之商为 $a_n2^{n-1}+a_{n-1}2^{n-2}+\cdots+a_12^0$，而余数为 a_0。

同样，上述商 $a_n2^{n-1}+a_{n-1}2^{n-2}+\cdots+a_12^0$ 若再除以 2，所得余数即为 a_1。依次类推，一直除下去，直到商为 0 为止，此时的余数就是 a_n。我们把这种将十进制整数转换成二进制数的方法称为“除 2 取余法”。

例如：将 25 转换为二进制数。

解

$25\div2=12$	余数	$1\rightarrow k_0$
$12\div2=6$		$0\rightarrow k_1$
$6\div2=3$		$0\rightarrow k_2$
$3\div2=1$		$1\rightarrow k_3$
$1\div2=0$		$1\rightarrow k_4$

所以 $(25)_{10}=k_4k_3k_2k_1k_0=(11001)_2$

4. 十进制小数转换成二进制小数（乘 2 取整法）

如果一个十进制小数 y 已被表示成一个二进制小数 $(0.a_{-1}a_{-2}\cdots a_{-m})_2$，那么 y 可按二进制数的权展开如下：

$y=a_{-1}2^{-1}+a_{-2}2^{-2}+\cdots+a_{-m}2^{-m}$

将等式两边同时乘 2，则得

$2\times y=a_{-1}+a_{-2}2^{-1}+\cdots+a_{-m}2^{-(m-1)}$

这里等号右边 a_{-1} 的取值非 0 即 1，而以下各项之和应为一纯小数。根据两个数相等的比较法则，不难理解 a_{-1} 的取值决定于 $2\times y$ 乘积的整数部分，即若 $2\times y$ 大于或等于 1，则 a_{-1} 取值为 1，否则取值为 0。同样，若将乘积的小数部分再乘以 2，可以得到 a_{-2} 的值。

依次类推可逐个求得 a_{-3}，…，a_{-m} 的值，直到乘积的小数部分为 0 时为止。

有的十进制小数在转换成二进制小数时，上面的计算过程可能会无限制地进行下去，此时可根据精度要求，选择适当的位数。我们把这种由十进制小数转换成二进制小数的方法称为“乘2取整法”。

例如：将0.90625转换成二进制数。

解

$0.90625\times2=1.8125$ 整数 $1\rightarrow k_{-1}$

$0.81250\times2=1.6250$ $1\rightarrow k_{-2}$

$0.6250\times2=1.2500$ $1\rightarrow k_{-3}$

$0.2500\times2=0.5000$ $0\rightarrow k_{-4}$

$0.5000\times2=1.0000$ $1\rightarrow k_{-5}$

所以 $(0.90625)_{10}=0.k_{-1}k_{-2}k_{-3}k_{-4}k_{-5}=(0.11101)_2$

对于具有整数和小数部分的十进制数，只要分别将整数部分和小数部分转换成二进制数，然后合并起来即可得到结果。

（二）十六进制数

1. 十六进制数的表示

十六进制数有两个主要特点：首先，它有16个不同的数字和字母符号0～9以及A、B、C、D、E、F，分别表示十进制数0～15；其次，数位之间是“逢十六进一”。因此，在不同的数位，数码所表示的值是不同的。

例如：$(4AC.5E)_{16}=4\times16^2+10\times16^1+12\times16^0+5\times16^{-1}+14\times16^{-2}=(1196.3672)_{10}$

由此可见，十六进制就是基数为16的进位计数制，它各数位上的位值（权）分别为$16^n\cdots16^1\cdots16^{-m}$。

2. 十六进制与二进制之间的转换

由于4位二进制数恰好表示16个数的组合，即一个十六进制数与4位二进制数是完全一一对应的，所以十六进制数与二进制数的转换，是十分简单的。

（1）十六进制转换成二进制

只要将每一位十六进制数用相应的4位二进制数替代即可。

例如：将$(5EA.7B4)_{16}$转换成二进制。

解

5	E	A	7	B	4
↓	↓	↓	↓	↓	↓
0101	1110	1010	0111	1011	0100

所以 $(5EA.7B4)_{16}=(101\ 1110\ 1010.0111\ 1011\ 01)_2$

由于在本例中二进制数最高位的一个0和小数点后最低的两个0无意义，所以最后从结果中将它们舍去不写。

（2）二进制转换成十六进制

以小数点为界，向左向右每4位为一组，依次写出每组4位二进制数所对应的十六进制数即可。

例如：二进制数$(1111111000111.100101011)_2$转换成十六进制数。

(000) 1	1111	1100	0111.	1001	0101	1 (000)
↓	↓	↓	↓	↓	↓	↓
1	F	C	7	9	5	8

结果，$(1111111000111.100101011)_2=(1FC7.958)_{16}$

转换时应注意最后一组若不足 4 位时须加“0”补齐 4 位。上例中最后一组为 1，若忘记补 0，可能视为十六进制数 01H，而事实上它应为十六进制数 08H。

对于十六进制数，常用加后缀 H 来表示，例如 64H 表示十进制数 100。

3. 十进制转换为十六进制

一般有下述两种方法：

(1) 直接法

这是一种类似于十进制数转换为二进制数的方法，即整数部分采用“除 16 取余法”，而小数部分采用“乘 16 取整法”。小数部分的转换通常会有误差，一般转换到所要求的精度为止。

(2) 间接法

这种方法是先将十进制数转换成二进制数，再将二进制数转换成十六进制数。

使用二进制数当数据较大时，由于位数太长，书写与阅读均很不便。因此，通常使用十六进制数来作为二进制数的缩写。在微型计算机中，目前通用的字长为 8 位，这恰可用 2 位十六进制数来表示，因此十六进制数应用十分普遍。

表 1-1 列出了一些十进制数和二进制数、十六进制数之间的对照关系。

表 1-1 十进制、二进制数、十六进制数对照表

十进制数	二进制数	十六制数	十进制数	二进制数	十六制数
0	0	0	14	1110	E
1	1	1	15	1111	F
2	10	2	16	10000	10
3	11	3	17	10001	11
4	100	4			
5	101	5	50	00110010	32
6	110	6	100	01100100	64
7	111	7	255	11111111	FF
8	1000	8			
9	1001	9			
10	1010	A	0.5	0.1	0.8
11	1011	B	0.25	0.01	0.4
12	1100	C	0.125	0.001	0.2
13	1101	D	0.0625	0.0001	0.1

二、二进制编码

在计算机中采用的是二进制数，因而数、字母、符号等在计算机中都要以特定的二进制码来表示，这就是二进制编码。

(一) 二进制编码的十进制数 (BCD 码)

1. 8421 BCD 码

在计算机中使用二进制代码进行工作，尽管它有许多机器设备方面的优点，但这就意味着使用它的人需要掌握并经常要进行从十进制数到二进制数的转换，才能让计算机接受，这是颇费功夫的。因此找到了一个比较适合于十进制系统的二进制代码的特殊形式：用 4 位二进制代码来表示 1 位十进制数，称之为十进制数的二进制代码表示法，简称 BCD 码。

由于 4 位二进制数从 0000 到 1111 可以表示 16 个数，所以理论上可以任选其中的 10 种来表示 0 到 9 的 10 个数字，但通常采用 0 到 9 各数字所对应的 8421 码作为其代码，从而称作 8421BCD 码。这种编码方式与十进制数的关系相当直观，它们之间的转换也是十分简单的，

见表 1-2。

表 1-2　8421 BCD 码

十进制数	8421 BCD 码	十进制数	8421 BCD 码
1	0001	7	0111
2	0010	8	1000
3	0011	9	1001
4	0100	10	0001 0000
5	0101	11	0001 0001
6	0110	12	0001 0010

例如：将十进制数 87.4 转换成 BCD 码。

解　$(87.4)_{10}=(1000\ 0111.0100)_{BCD}$

又如：将 BCD 码 0110 1001.0101 0010 转换为十进制数。

解　$(0110\ 1001.0101\ 0010)_{BCD}=(69.52)_{10}$

必须指出：BCD 码与真正的二进制数是不同的，它形式上为二进制数，实际上为十进制数。BCD 码数常用在微机输入输出设备中，作为机器的二进制与人们习惯使用的十进制之间一种过渡性的编码，以简化人机联系。

2. 二-十进制调整

由于计算机总是将数作为二进制数来处理的，而 BCD 码数是形式上的二进制数，实际上为十进制数，它要求“逢十进一”，因此若将这种 BCD 码数交给计算机直接进行运算，结果可能出错。

例如：求 BCD 码 5+8 的结果。

解
```
  0101      5 的 BCD 码
+1000       8 的 BCD 码
------
  1101
```

1101 为非法 BCD 码，说明结果出错。

例如：求 BCD 码 9+7 的结果。

解
```
  1001      9 的 BCD 码
+0111       7 的 BCD 码
------
 10000
```

运算结果 $(10000)_{BCD}$应为 10 的 BCD 码，可见这两种情况均使结果发生了错误。

为了得到正确的 BCD 码运算的结果，必须进行二-十进制调整，规则如下：

① 4 位（二进制数）一组，若两个 BCD 数相加结果大于 9 时，则应对该 4 位进行“加 6 调整”。

② 4 位（二进制数）一组，若两个 BCD 数相加结果等于或大于 16 时，则应对该 4 位进行“加 6 调整”。

读者可以根据这两条规则对上述两例的结果进行调整，以便得出正确的 BCD 码运算结果。

在计算机指令系统中，有一条专门用于 BCD 码调整的指令（在 MCS-51 指令系统中为 DAA），使用起来十分方便。

（二）字符的 ASCII 码

在计算机中除数字用二进制表示外，字母和各种字符也必须用二进制表示，目前最普遍使用的为 ASCII 码。ASCII 码是美国信息交换标准代码（American Standard Code for Information Interchange）的缩写，它采用 7 位二进制代码来对字符进行编码，故可表示 128 个不同的字符，见附录 A。

阿拉伯数字 0～9 的 ASCII 码分别为 30H～39H，大写英文字母 A、B、…、Z 的 ASCII 码则是从 41H 开始依次往下编排。回车符 CR 的 ASCII 码为 0DH。

三、带符号数的表示

在计算机中，所有的数据、指令以及符号都是用二进制代码表示的。

我们把一个数在计算机中被表示的二进制形式称为机器数，而这个数则称为该机器数的真值。机器数具有下列特点：

① 为便于操作，并考虑到计算机设备上的限制，机器数有固定的位数。因此机器的真值受到固定位数的限制，具有一定的范围，超过了这个范围就会产生“溢出”。一个 8 位机器数所能表示的无符号整数的范围为 00000000B～11111111B，即十进制数 0～255，超过这个范围就会产生“溢出”。

② 机器数把其真值的符号数字化。通常用其最高位作为符号位，最高位取 0 或 1 表示其真值为正或负。一个 8 位二进制数所能表示的带符号数的范围，在定点原码表示情况下为 −127～+127，超出这个范围也会发生“溢出”。

③ 将一个 8 位机器数看作无符号数还是带符号数，这完全是程序设计人员在事先约定的。

④ 在机器数中，采用定点或浮点方式来表示小数点的位置。

（一）原码、反码、补码

（1）原码

整数 x 的原码是指：其符号位的 0 或 1 表示 x 的正负，其数值部分就是 x 绝对值的二进制表示。通常用 $〔x〕_{原}$ 表示 x 的原码。

设 $x = x_1x_2\cdots x_{n-1}$，其中 x_i 为一位二进制数，i=1、…、(n−1)

则 $〔x〕_{原}=\begin{cases}0\ x_1x_2\cdots x_{n-1} & \text{当 } x \geqslant 0 \text{ 时}\\ 1\ x_1x_2\cdots x_{n-1} & \text{当 } x < 0 \text{ 时}\end{cases}$

例如：设机器数的位数是 8，其中最高位为符号位，其余是数值部分，那么：

$〔+17〕_{原}=00010001$，　　$〔-39〕_{原}=10100111$。

综上所述，不难证明数 0 的原码有两种不同的形式：

$〔+0〕_{原}=00000000$，　　$〔-0〕_{原}=10000000$。

8 位二进制数用原码表示的数的范围为−127～+127。

在进行带符号数运算时，数的原码表示显得很不方便，于是引进了数的反码表示法。

（2）反码

整数 x 的反码是指：对于正数，反码与原码相同，即其符号位为 0，其余位为其数值位本身；对于负数，反码的符号位为 1，其余位为数值位按位取反（即将 1 变为 0，将 0 变为 1）。通常用 $〔x〕_{反}$ 表示 x 的反码。

设 $x = x_1x_2\cdots x_{n-1}$，其中 x_i 为一位二进制数，$i = 1、\cdots、(n-1)$

则 $[x]_{反}=\begin{cases}0\ x_1x_2\cdots x_{n-1} & \text{当 } x\geqslant 0 \text{ 时}\\ 1\ \overline{x_1}\ \overline{x_2}\cdots\overline{x_{n-1}} & \text{当 } x<0 \text{ 时}\end{cases}$

例如：设机器数的位数是 8，其中最高位为符号位，其余是数值部分，那么：

$[+17]_{反}=[+17]_{原}=00010001$，　　$[-39]_{反}=11011000$。

综上所述，不难证明数 0 的反码有两种不同的形式：

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

8 位二进制数用反码表示的数的范围也为－127～＋127。

负数的反码表示也不能适应计算机算术运算的要求，因此它仅作为求取负数补码的中间过程而存在。

(3) 补码

整数 x 的补码可用下面方法求得：正数的补码与其原码相同；负数的补码是将其原码除符号位外的各位取反（即 0 变 1，1 变 0），然后在最低位上加 1。通常用 $[x]_{补}$ 表示 x 的补码，用 $\overline{x_i}$ 表示对 x_i 位取反。

设 $x=x_1x_2\cdots x_{n-1}$，其中 x_i 为一位二进制数，$i=1、\cdots、(n-1)$

则 $[x]_{补}=\begin{cases}0\ x_1x_2\cdots x_{n-1}=[x]_{原} & \text{当 } x\geqslant 0 \text{ 时}\\ 1\ \overline{x_1}\ \overline{x_2}\cdots\overline{x_{n-1}}+1 & \text{当 } x<0 \text{ 时}\end{cases}$

例如：$[+14]_{补}=[+14]_{原}=00001110$

又如：求 $[-36]_{补}$

由于 $[-36]_{原}=10100100$，故 $[-36]_{反}=11011011$，因此 $[-36]_{补}=11011100$

数 0 的补码表示是唯一的，即 $[0]_{补}=[+0]_{补}=[-0]_{补}=00000000$

8 位二进制数用补码表示的数的范围为－128～＋127。

对于补码来说，负数补码须再求补以后才能得到其真值，即 $[(x)_{补}]_{补}=[x]_{原}$。这也就是说，对于一个用补码表示的二进制数，其最高位为符号位，当符号位为 0（即正数）时，其余 7 位即为此数的二进制值，但当符号位为 1（即负数）时，其余 7 位不是此数的二进制值，把它们按位取反，且在最低位加 1 后方为它的二进制值。

例如：已知 $[x]_{补}=10010100$，此时 x 的真值并不等于 $(-20)_{10}$，而应有：

$x_{真值}=-1101100=(-108)_{10}$

(4) 求补

已知一个数的补码，求其相反数的补码的运算称为求补。求补的方法为：连同符号位在内，一起求反加 1。

例如：已知 $[x]_{补}=11111100$（－4 的补码），求 $[-x]_{补}$。

解　$[-x]_{补}=00000100$(＋4 的补码)

(二) 补码在运算中的应用

若有 x、y 两个数相加，那么应根据 x 和 y 的正负号来决定做加法或减法。但是，采用补码以后，由于 $[x+y]_{补}=[x]_{补}+[y]_{补}$，所以只要将 x 的补码和 y 的补码相加，而不必去考虑它们的正负号，就能得到正确的两数和的补码。

由此可见，采用补码以后，可使正、负数的加、减运算简化为单纯的相加运算，这就是引入补码概念的目的所在。

若有 x、y 两个数相减，那么应根据 x 和 y 的正负号而决定做加法或做减法。但采用补码以