高等学校规划教材

微型计算机原理与接口技术

吴永祥 主编

煤炭工业出版社

高等学校规划教材

微型计算机原理与接口技术

吴永祥　刘万军
傅国军　傅兴武　胡业林　编

煤炭工业出版社

（京）新登字042号

内 容 提 要

本书是煤炭系统部分高校工业自动化专业综合改革的配套教材。全书共三篇，第一篇以Z80为核心介绍微型计算机基础知识、微处理器结构、存储器、指令系统、汇编语言程序设计及输入、输出和中断；第二篇介绍MCS-51和MCS-96系列单片微型计算机的基本组成原理、指令系统及程序设计；第三篇介绍微型计算机接口技术及应用，包括微型计算机的总线、存储器扩展、并行串行接口扩展、A/D和D/A转换、接口设计及应用。

本书可作为大专院校微型计算机原理与接口技术课程的教材或参考书，也可供应用微型计算机的科技人员学习和参考。

高等学校规划教材

微型计算机原理与接口技术

吴永祥　刘万军
傅国军　傅兴武　胡业林　编

责任编辑：高　专

*

煤炭工业出版社　出版
（北京安定门外和平里北街21号）
煤炭工业出版社印刷厂　印刷
新华书店北京发行所　发行

*

开本 787×1092mm　1/16　印张 22.5
字数 531千字　印数 1—2 700
1995年10月第1版　1995年10月第1次印刷
ISBN 7-5020-1242-7/TP32
书号 **4010·A0323**　定价：**18.00**元

出 版 说 明

《中国教育改革和发展纲要》指出，教育改革“要按照现代科学技术文化发展的新成果和社会主义现代化建设的实际需要，更新教学内容，调整课程结构，加强基本知识、基础理论和基本技能的培养和训练，重视培养学生分析问题和解决问题的能力”。“高等学校教材要在积极扩大种类的同时，不断提高质量，加强理论与实际的联系，力求思想性和科学性的统一”，要适应教学改革需要。

由阜新矿业学院、山西矿业学院、黑龙江矿业学院、淮南矿业学院等四所院校进行的工业电气自动化专业教学综合改革，按照“知识归类、科学组合、优化设课、精简学时、加强实践”的改革原则，制定了新的教学计划。根据培养目标的要求，新计划对于原有课程采用删、调、合、增的方法改变原有课程体系，将工业电气自动化专业电类课程归并为四大类：基础理论课、方法论课、应用技术基础课和有针对性的专业课。

为了适应深化教学改革的需要，我们首先组织编写、出版以下几种教材：

电工原理

网络与系统分析

电机与传动

控制工程技术

微型计算机原理与接口技术

微型计算机测控技术

矿山自动控制系统

电工技术 CAA

电气实验技术

电气工程实践教程

煤炭工业部科技教育司

教材编审室

前　言

本书根据煤炭工业部部分高校工业自动化专业综合改革会议确定的关于“微型计算机原理与接口技术”教材编写指导意见，并在近几年来从事微型计算机原理及应用、微型计算机接口技术和单片微型计算机原理及应用课程教学和科研的基础上编写的。

本书以Z80为例较全面地介绍了计算机基础知识、单片微型计算机软件硬件及其应用、单片微型计算机接口技术等。本书概念清楚，内容丰富，讲解透彻，使读者通过对本书的学习，既掌握一定的计算机基础知识，又掌握单片微型计算机原理和接口技术。

本书讲授时间按90学时安排，在讲授时可以根据具体情况对有些内容酌情增减。

本教材由吴永祥任主编并编写第五章，第六章第一、二、四节；刘万军任副主编并编写第八、十二章；傅国军编写第一、二、九章，第十五章第二节；傅兴武编写第六章第三节，第十、十一、十三章；胡业林编写第三、四、七章，第十五章第一节。

在编写过程中，得到煤炭工业部科教司、教材编审室及部分高校领导和有关人员的关心、支持和帮助。在此谨致忠心的谢意。

由于作者水平有限，编写时间仓促，书中难免存在错误与不足之处，殷切希望广大读者批评指正。

编　者

一九九四年十二月

目　　录

绪　论

数字电子计算机的出现是近代重大科学成就之一。自从1946年第一台数字电子计算机诞生至今只不过40多年的历史，但是它的发展速度却极为惊人，应用也相当广泛。电子数字计算机（简称为计算机）已渗透到国防尖端、工矿企业、农业、企业管理、交通运输、日常生活的各个领域，其作用和成就日益卓著，成为现代工业水平的标志之一。

40多年来，计算机的发展已经历了四代，目前正在向第五代发展。

从1946年到1957年为第一代。第一代为电子管数字计算机。此时，计算机的逻辑元件采用电子管，主存储器采用磁芯、磁鼓，外存储器开始采用磁带机；软件主要使用机器语言，这时，符号语言已经出现并开始使用，运算速度为每秒几千次到几万次。用途主要是科学计算。

从1957年到1964年为第二代。第二代计算机的逻辑元件开始采用晶件管，主要存储器仍为磁芯，但外存储器开始使用磁盘；软件有了很大发展，高级语言已普遍使用，操作系统随之出现。计算机运算速度也相应提高，每秒可达几万次到几十万次。应用范围已扩展到各种事务数据处理，并开始应用于过程控制。

从1964年到1970年为第三代。第三代计算机开始采用中小规模集成电路。比如，逻辑元件等。此时，主存储器虽然仍为磁芯，但是计算机种类已发展为多样化、系列化，外部设备不断增加，种类繁多。软件、操作系统进一步完善，分时系统、多道程序都有所发展并广泛应用。此外，在发展大型计算机的同时，小型和超小型计算机在迅速发展，其运算速度达每秒几十万次至几百万次，广泛地应用于科学计算、工业控制、数据处理等领域。

第四代计算机是指全面采用大规模集成电路的机器。从70年开始，第三代计算机逐步发展和过渡为第四代计算机。到1975年研制成功的470V/6型计算机和M-190计算机都是大规模集成电路的大型机，可以作为第四代计算机的代表机型。第五代计算机是智能计算机，功能不仅可以模拟人的神经而且具有学习功能；软件将发展为自然语言。

计算机目前正在朝两个方向发展：一是大型、巨型化，二是小型和微型化。

微型计算机（Microcomputer）与其他计算机的区别在于它的中央处理单元（CPU）是采用大规模集成技术集成在一块芯片上，而其他的大、中、小型计算机的CPU是由多片集成电路组成。为了和其他计算机的CPU相区别，称微型计算机的CPU芯片为微处理器。

自从1971年美国Intel公司研制成功以4004微处理器为核心的4位微计算机以来，短短20年里，微机得到了迅猛的发展。微处理器的集成度差不多隔两年就翻一番，且性能也能增长一个数量级。至今微处理器及其微型计算机也经历了四代的演变。

从1971年至1972年为第一代。第一代微处理器的代表产品是美国Intel公司首先研制成功的4004 4位微处理器。4位微处理器虽然性能较简单，但由于价格便宜，在玩具、计算器、游戏机以及其他简单控制场合仍有广泛的应用。同年，该公司还研制出8008，它是最早的8位微处理器。

从1973年至1977年为第二代。第二代的代表产品是美国Intel公司的8080、Motorola

公司的 6800 和 Zilog 公司的 Z80，它是 8 位微处理器。在这一期间，还出现了 8 位单片微型计算机(简称为单片机)，它是将一台微型计算机的最基本部件，包括微处理器、存储器和一部分接口电路都集成在一块芯片上而成的。1976 年问世的 8 位单片机 8048 是 Intel 公司的早期产品。

从 1978 年至 1981 年为第三代。这一代的代表产品是美国 Intel 公司的 8086、Zilog 公司的 Z8000 和 Motorola 公司的 M68000，它们是 16 位微处理器。

1981 年以后为第四代。这一代的代表产品是美国 Intel 公司的 Iapx432、BELL 研究所的 MAC-32 和 NS 公司的 NS16032，它们是 32 位微处理器，属于超大规模集成电路。

在微处理器由低档向高档发展的同时，单片机的发展也十分迅速。在单片机的发展史上，美国的 Intel 公司贡献较为突出。自 1976 年，该公司研制出第一代 8 位通用单片机——MCS-48 系列，1980 年推出第二代 8 位增强型单片机——MCS-51 系列，1983 年又推出了 16 位单片机——MCS-96 系列，近年 80960 32 位单片机已经投入市场并开发应用。

包括单片机在内的微型计算机应用是非常广泛的。其主要应用有如下几个方面：

1. 用于科学计算。计算机由于运算速度快是其他工具无法比拟的，于是在科学计算中发挥了强大威力。

2. 用于数据处理。在企业管理、会计、统计、资料和实验资料整理等数据加工、合并、分类工作中用计算机管理不仅方便，而且还能大大提高管理水平。

3. 用于过程控制。利用计算机进行生产过程的实时控制，可以大大提高自动化水平，提高控制的准确度，提高产品质量。近年来，微机控制在电力、矿业、石油、机械、化工、铁路运输及轻工业等许多部门都得到广泛的应用。

4. 用于机、电、仪一体的智能产品。微型计算机尤其单片机具有体积小、功耗低、控制能力强优点，能把它做到产品内部，取代部分老式机械、电子零件和元器件，可以使产品缩小体积、增强功能、实现智能化。如电子称、银行计息电脑、录音机芯、微型打印机等产品都是单片机的具体应用。

5. 用于计算机辅助设计。辅助设计简称为 CAD，它是利用计算机的功能部分地代替人工来进行各种产品的分析和设计。从而缩短设计周期，提高设计成功率和可靠性。例如，在大规模集成电路设计、机械设计和建筑设计中均有应用。

6. 用于通信。计算机技术和通信技术相结合构成的计算机通信网是现代化通信的重要手段。

完全可以预料，像人类赖以拓展自身功能的其它工具一样，微型计算机一定会在人类认识世界和改造世界的过程中，辅助人们的各种社会活动，并在该过程中被人类开发得更加造价低廉、性能完美、易于维护，成为人类不可缺少的一种智能工具。

第一篇　微型计算机基础

第一章　微型计算机运算基础

在数字计算机中常使用的数制是二进制、十进制和十六进制。十进制是为了让人们便于识别。而计算机本身则采用二进制和十六进制表示各种数字、图形和符号。本章主要介绍数在计算机中的表示、运算及各种常用的编码方法。

第一节　数制和编码

计算机最基本的功能是对数据进行运算和处理，数在计算机中则用电平高低等物理状态来表示。计算机中一般采用二进制数，原因是它只要求两个不同的物理状态。而为了简便和直观，在对微机中的数进行分析讨论时，则常用十六进制数。本节将对数制与码制问题进行简要分析。

一、进位计数制

按进位的方法进行计数称进位计数制。常用的进位计数制有以下几种。

1. 十进制

十进制是人们最习惯的进位制。它只有 0，1，2，…，9 共 10 个不同的数字；采用“逢十进一”的进位法则，即相邻位的“权”相差十倍。例如

$$123.45=1\times10^{2}+2\times10^{1}+3\times10^{0}+4\times10^{-1}+5\times10^{-2}$$

一般地，任意一个十进制数 D 为

$$\begin{aligned}D=&D_{n-1}\times10^{n-1}+D_{n-2}\times10^{n-2}+\cdots+D_{1}\times10^{1}+D_{0}\times10^{0}+D_{-1}\times10^{-1}+\\&+D_{-2}\times10^{-2}+\cdots+D_{-m}\times10^{-m}\\=&\sum_{i=-m}^{n-1}D_{i}\times10^{i}\end{aligned}$$

其中，D_i 表示第 i 位数码，可以是 0～9 中的任一数，其值取决于数值 D；m、n 为正整数，n 为小数点左边的位数，m 为小数点右边的位数。10 为十进制数的基数（或称底数），而 10^{n-1}、…10^{0}、…10^{-m}等则分别为该位的权。

2. 二进制

二进制数是计算机中最常用的，它仅有 0 和 1 两个数码，进位法则是“逢二进一”。例如

$$(1011.101)_2=1\times2^{3}+0\times2^{2}+1\times2^{1}+1\times2^{0}+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}=(11.625)_{10}$$

其中，（ ）$_2$ 与（ ）$_{10}$分别代表二进制数与十进制数。

一般的二进制数 B 可表示为

$$\begin{aligned}B=&B_{n-1}\times2^{n-1}+B_{n-2}\times2^{n-2}+\cdots+B_{1}\times2^{1}+B_{0}\times2^{0}+B_{-1}\times2^{-1}+\\&+B_{-2}\times2^{-2}+\cdots+B_{-m}\times2^{-m}\\=&\sum_{i=-m}^{n-1}B_{i}\times2^{i}\end{aligned}$$

其中，B_i 只能取 0 或 1，其值取决于数值 B；m、n 为正整数；二进制的基数为 2。

3. 十六进制

十六进制数常用来表示数据，它有 16 个不同的数码，即 0～9、A、B、C、D、E、F，其中 A～F 分别等于十进制数 10～15，见表 1-1。十六进制数的进位法则为“逢十六进一”。例如

$$(2A3F)_{16}=2\times16^3+10\times16^2+3\times16^1+15\times16^0=(10815)_{10}$$

一般的 16 进制数 H 可表示为

$$H=H_{n-1}\times16^{n-1}+H_{n-2}\times16^{n-2}+\cdots+H_1\times16^1+H_0\times16^0+H_{-1}\times16^{-1}+$$
$$+H_{-2}\times16^{-2}+\cdots+H_{-m}\times16^{-m}$$
$$=\sum_{i=-m}^{n-1}H_i\times16^i$$

其中 H_i 可取 0～F 间的任一值取决于数值 H；m、n 为正整数，基数为 16，故称十六进制数。

表 1-1　二进制、十进制与十六进制数码对照表

十进制数	十六进制数	二进制数	十进制数	十六进制	二进制数
0	0	0 0 0 0	9	9	1 0 0 1
1	1	0 0 0 1	10	A	1 0 1 0
2	2	0 0 1 0	11	B	1 0 1 1
3	3	0 0 1 1	12	C	1 1 0 0
4	4	0 1 0 0	13	D	1 1 0 1
5	5	0 1 0 1	14	E	1 1 1 0
6	6	0 1 1 0	15	F	1 1 1 1
7	7	0 1 1 1	16	10	10 0 0 0
8	8	1 0 0 0			

一般情况，数可写作

$$K=\sum_{i=-m}^{n-1}K_iJ^i$$

其中，J 为基数；K_i 为第 i 位数码，可取 0～（J－1）中的任一值，取决于数 K；J^i 为 K_i 的“权”；m、n 为正整数，n 为小数点左边位数，m 为小数点右边位数。

通常用 $(K)_J$ 表示 J 进位制数。在微机中也常用某些字母来代表不同的进位制数。例如十进制数的代号为 D（Deci-mal）或省略；二进制数代号为 B（Binery）或‰；八进制数代号为 Q 或 O（Octal），十六进制数代号为 H（Hexadecimal）或 $。

二、数制转换

1. 二进制数转换为十进制数

根据其定义，可将二进制数按权展开后相加，如

$$(101.001)_2=1\times2^2+1\times2^0+1\times2^{-3}=(5.125)_{10}$$

2. 十进制数转换为二进制数

我们先以 $(37)_{10}$ 为例，分析整数的变换，即

$$(37)_{10}=(K_{n-1}\cdots K_2K_1K_0)_2=K_{n-1}\times2^{n-1}\cdots+K_1\times2^1+K_0\times2^0$$
$$=2(K_{n-1}\times2^{n-2}+\cdots+K_1\times2^0)+K_0$$

等式两端同除以 2，得

$$(18)+1/2=(K_{n-1}\times2^{n-2}+\cdots+K_1\times2^0)+K_0/2$$

等式两端括弧内的整数部分与其余的小数部分应分别相等，故 $K_0=1$，即 37 除以 2 后的余数为 K_0，这样就分离出 K_0。再除以 2，即可分离出 K_1，依此类推，可得其余的 $K_2\cdots K_{n-1}$。整个计算过程为

2	37	……余 $1=K_0$　（最低位）
2	18	……余 $0=K_1$
2	9	……余 $1=K_2$
2	4	……余 $0=K_3$
2	2	……余 $0=K_4$
	1	……余 $1=K_5$　（最高位）

则　　$(37)_{10}=(K_5K_4K_3K_2K_1K_0)_2=(100101)_2$

我们再以 $(0.625)_{10}$为例分析小数的变换，如

$$(0.625)_{10}=(0.K_{-1}K_{-2}\cdots K_{-m})=K_{-1}\times 2^{-1}+K_{-2}\times 2^{-2}+\cdots+K_{-m}\times 2^{-m}$$

等式两端同乘以 2 得

$$(1+0.25)_{10}=K_{-1}+(K_{-2}\times 2^{-1}+\cdots+K_{-m}\times 2^{-m+1})$$

由等式两端的整数部分相等，可分离出 $K_{-1}=1$。再乘以 2，可分离出 K_{-2}，依此类推，可得其余数码。整个转换过程为

0.625	
× 2	
1.250	……整数部分$=1=K_{-1}$　（最高位）
0.250	（取小数部分）
× 2	
0.500	……整数部分$=0=K_{-2}$
× 2	
1.000	……整数部分$=1=K_{-3}$　（最低位）

则　　$(0.625)_{10}=(0.K_{-1}K_{-2}K_{-3})=(0.101)$

有的数的计算过程会无限地进行下去，这时可根据精度要求选取适当的位数。

总之，将十进制数转换为二进制数（即十换二）时，其整数部分不断除以 2，每次所得余数即为二进制数码，最后余数为最高有效位的数，称除 2 取余法；而小数部分则不断乘以 2，每次乘积的整数部分依次排列成二进制数码，称乘 2 取整法。

3. 十六进制与十进制间的转换

（1）十六进制转换十进制时将十六进制数按权展开，如

$$(F4)_{16}=F4H=15\times 16^1+4\times 16^0=(244)_{10}=244D=244$$

（2）十进制转换十六进制与十进制转换二进制类似，整数部分用除 16 取余法，小数部分用乘 16 取整法。如 $(1200.28125)_{10}=(?)_{16}$换算如下：

16	1200	……余 $0=K_0$
16	75	……余 $11=BH=K_1$
		……余 $4=K_2$

而

```
      0.28125
  ×        16
-----------------
     168750
+     28125
-----------------
     4.50000   ……整数部分=4=K-1
     0.50000     (取小数部分)
  ×        16
-----------------
     8.00000   ……整数部分=B=K-2
```

所以 $(1200.28125)_{10}=(4B0.48)_{16}$

4. 十六进制与二进制间的转换

在讨论微型机的数据时，经常使用十六进制数，由于 $2^4=16$，可用 4 位二进制数表示 1 位十六进制数见表 1-1，这两种数制间可以很方便地进行转换。对于十六进制转换二进制，可将每位十六进制数用相应的 4 位二进制数代替，如

$$(04B2)_{16}=(000\ 0100\ 1011\ 0010)_2=(10010110010)_2$$

反过来，对二进制转换十六进制则将二进制数以小数为准，分别向左和向右按 4 位一组转换为对应的十六进制数，如 $(1110\ \ 0011.\ 1010\ \ 0111)_2=(E3\cdot A7)_{16}$。

三、计算机中数的表示法

1. 机器数真值

在数学中，带符号数的正、负可分别用符号“+”、“−”来表示。而计算机里的电子器件一般只有“0”和“1”两种状态，因此数的符号也可用这两种数码来表示，在数的前面增设一个符号位。正数的符号位为“0”，负数的符号位为“1”。

例如，有两个数

$$X=+1010100B\quad Y=-1010100B$$

在计算机中可以表示为

$$X1=01010100B\quad Y1=11010100B$$

这种数码化的带符号数就称为机器数，而它们所代表的原来的数则称为真值。机器数一般是计算机里存储或进行运算的数码；真值则是数学上人们习惯使用的带符号数，即机器数所代表的真正数值。上例中的 X1、Y1 即为机器数，而 X、Y 则分别是它们的真值。机器数通常有原码、反码和补码三种表示方法。

2. 原码表示

如上所述，正数的符号位用 0 表示，负数的符号位用 1 表示，其余各位为数值位，即为该数的绝对值。这种数的表示方法称为原码表示法。在前述例子中，X1 与 Y1 即为原码表示，可写作

$$X=+1010100B=+84\quad [X]_{原}=X1=01010100B$$
$$Y=-1010100B=-84\quad [Y]_{原}=Y1=11010100B$$

0 的原码表示可以是（+0），也可认为是（−0），即 0 在原码中有两种表示法 $[+0]_{原}=000\cdots0$、$[-0]_{原}=100\cdots0$。

在表 1-2 中列出了 8 位二进制数的各种表示法。最左边一栏为机器数，右边四栏为各种

表示方法所对应的真值。由表可见，8 位二进制数的原码表示范围为－127～＋127。

原码表示简单易懂，与真值转换方便。但同号数相减或异号数相加时要进行减法运算。因此，引出了数的反码和补码表示法。

表 1-2　8 位二进制数的表示法

二进制数码表示	符号二进制数	原　码	反　码	补　码
00000000	0	＋0	＋0	＋0
00000001	1	＋1	＋1	＋1
00000010	2	＋2	＋2	＋2
⋮	⋮	⋮	⋯	⋮
01111101	125	＋125	＋125	＋125
01111110	126	＋126	＋126	＋126
01111111	127	＋127	＋127	＋127
10000000	128	－0	－127	－128
10000001	129	－1	－126	－127
10000010	130	－2	－125	－126
⋮	⋮	⋮	⋮	⋮
11111101	253	－125	－2	－3
11111110	254	－126	－1	－2
11111111	255	－127	－0	－1

3. 反码表示

正数的反码表示与原码相同，最高位为符号位，此位为 0 表示其为正数，其余位为数值位。如 X＝＋1010100B，$[X]_{反}=[X]_{原}=01010100B$

负数的反码表示其符号位为 1，数值位取反，即将正数数值位中的 0 变 1，1 变 0。或者说将正数的各位全部取反（包括符号位）即为负数的反码表示。如

$$X=-1010100B \quad [X]_{反}=10101011B$$

0 的反码表示有两种：$[+0]_{反}=000\cdots0$　$[-0]_{反}=111\cdots1$

8 位二进制数的反码表示范围为－127～＋127 见表 1-2。

4. 补码表示

正数的补码表示与原码和反码相同，而负数的补码表示为其反码的最低位加 1。如 X＝－1010100B　$[X]_{反}=10101011B$　$[X]_{补}$ 10101100B，0 的补码表示只有一种，即

$$[+0]_{补}=00\cdots0B \quad [-0]_{补}=[-0]_{反}+1=11\cdots1B+1=00\cdots0B$$

由表 1-2 可见，8 位二进制数的补码表示范围为－128～＋127。由于补码表示中 0 只占一个机器数，故其范围比原码和反码扩大了（由－127 扩大为－128）。

实际上，补码的概念可通过日常生活现象来理解。如果钟表由 9 点拨到 5 点，可以倒拨四格，即 9－5＝4。但有的钟表不允许倒拨，就只有顺拨 7 格，也指向 5 点，当到达 12 点时，即由 0 重新开始，相当于自动丢失了 12。即 9＋7＝12（自动丢失）＋4＝4 这个自动丢失的数（12）称为模（modulo，简写为 mod），是此系统的量程或系统内所能表示的最大数。对模 12，可写作 9－5＝9＋7（mod 12）表明等号两边同除以模 12，其余数相同，即

(9－5) 与 (9＋7) 对模 12 是同余的，也可说 (－5) 与 (＋7) 对模 12 互为补数。

一般地，若数 X、Y、K 满足下式 Y＝nK＋X (n 为整数) 则称 Y 与 X 对模 K 同余，或互为补数，也可写作

$$Y \equiv X \pmod{K}$$

可见，当某数减去小于模之另一数时 (上例为 9－5)，可用加上此数的补数 (上例为 9＋7) 来代替。而补数就是模与该数的负数之和，因此，利用补数的概念和补码表示法就可以将减法运算转换为加法运算。其推导过程为

设二进制数共有 n 位，其模为 2^n，负数〔－X〕的反码表示为

$$〔-X〕_{反}=\underbrace{111\cdots1}_{n位}-X=(2^n-1)-X$$

而〔－X〕的补码表示则为

$$〔-X〕_{补}=〔-X〕_{反}+1=2^n-1-X+1=2^n-X$$

这就说明，以 2^n 为模，$〔-X〕_{补}$ 与－X 互为补数，减去 X 的运算可用加上 $〔-X〕_{补}$ 来实现。这种以 2^n 为模的补码也简称为以 2 为模的补码，而反码可看作以 (2^n-1) 为模的补码，简称对 1 的补码。而负数用补码表示时，也可用 (2^n-X) 来求取。例如

$$X=01010100B \quad 〔-X〕_{补}=2^8-X=100000000B-01010100B=10101100B$$

字长为 n 的机器只能表示 n 位数，因此，2^n (它是一个 n＋1 位数 100…0) 在机器中只能以 n 个 0 来表示，或者说 2^n 和 0 在机器中的表示形式是一样的。

如果将 n 位字长的存储单元的最高位留作符号位，对于正数和负数的补码为

$$〔X〕_{补}=\begin{cases} X & 2^{n-1}>X\geqslant 0 \\ |\overline{X}|+1 & 0>X\leqslant -2^{n-1} \end{cases}$$

即：正二进制数的补码与原码相同，负二进制数的补码等于它的绝对值按位取反加 1。

例 1-1　求出 X＝－87 的 二进制补码。

解：

$$〔X〕_2=-01010111B$$

$$〔X〕_{补}=|\overline{X}|+1=10101000+1B=10101001B$$

已知原码可依上述方法求出其补码，已知补码，只要对其再次求补就可得到原码。对正数来讲，两者相等。对负数的补码经再次的求补加上负号，就是其原值。

例如：对 $〔X〕_{补}=10110$，则 $X=-〔〔X〕_{补}〕_{补}=-01010$。

5. 补码运算

在微型计算机中，由于补码的加减运算比原码的运算简单，它是符号位与数值部分一起参加运算，并且能自动获得结果 (包括符号和数值都在内)。因此，带符号数一般都以补码的形式在机器中存放和进行运算。

设 X 和 Y 是两个正数，可以证明两个数和的补码等于两个数补码的和：

$$〔X+Y〕_{补}=2^n+〔X+Y〕=〔2^n+X〕+〔2^n+Y〕=〔X〕_{补}+〔Y〕_{补}$$

也可以证明这两个数差的补码等于减数的补码与被减数负值补码的和：

$$〔X-Y〕_{补}=2^n+〔X-Y〕=2^n+X+2^n+〔-Y〕=〔X〕_{补}+〔-Y〕_{补}$$

可见，在补码运算中，减法运算可以用加法来代替。两个数的补码运算可按下列步骤进行：

(1) 把参与运算的两个数连同其前面的正负号变成补码；

(2) 进行补码加法，得到两个数运算结果的补码，最高位上有进位则余弃不要；

(3) 如要运算结果的真值，则按补码求真值的办法进行。

例 1-2　用补码运算求出 64－10。

解：

(1) $〔64〕_{补}=01000000B$　$〔-10〕_{补}=11110110B$

(2) 01000000＋11110110＝00110110B

(3) 求真值，因数为正值，可直接求 $〔0011011\ 0〕_2=54$

例 1-3　用补码运算求出 64－65。

解：

(1) $〔64〕_{补}=01000000B$　$〔65〕_{补}=10111111B$

(2) 相加：01000000＋10111111＝11111111B

(3) 求真值：由于结果为负数，要求出其原码后再求真值

$〔11111111〕_{补}$ 的原码为 $〔-00000001〕_2$，其真值为－1

在计算机中，带符号数都以补码数在机器中存放，它们可以进行加法运算，也可以进行减法运算。但在实际机器中只有加法运算，减法也是通过加法来实现的。

由于补码的加、减法运算十分方便，一般微机中的带符号数除特别指明外，都采用补码表示。

四、编码

如前所述，计算机只能接受和处理二进制数，而实际要求计算机处理的信息往往需要十进制数、字母、符号、以及汉字或图形等。这些信息都采用二进制编码，然后才能送入计算机进行处理。

1. 二进制编码的十进制数（二-十进制码，BCD 码）

当要求计算机处理十进制数时，先将十进制数转换为二进制数，在计算机中对二进制数进行计算，计算结果再转换成十进制数输出，这个过程十分繁杂。为了简化处理过程，一些简单运算可直接采用二进制编码的十进制数，例如数字钟等。

用 4 位二进制数可表示 1 位十进制数，其编码方法有多种，常用 8421-BCD 码 (Binery Coded Decimal，二进制编码的十进制数)。表 1-3 列出了这一编码系统的表示方法，这实际上是将十六进制的最后 6 个状态去掉，仅保留前 10 个状态的结果。

表 1-3　8421BCD 编 码 表

十进制数	8421 BCD 码	十进制数	8421 BCD 码
0	0 0 0 0	6	0 1 1 0
1	0 0 0 1	7	0 1 1 1
2	0 0 1 0	8	1 0 0 0
3	0 0 1 1	9	1 0 0 1
4	0 1 0 0	10	0 0 0 1 0 0 0 0
5	0 1 0 1	11	0 0 0 1 0 0 0 1

BCD 码比较符合人们的习惯，可直接将它们按十进制位进行转换，例如 $(0010\ 1001\ 0101.\ 0010\ 1000)^{BCD}=295.28D$，$347D=(0011\ 0100\ 0111)^{BCD}$

2. 字符的编码