高等学校教材

微型计算机原理与应用

8051、8098 单片机应用基础

WEIXING JISUANJI YUANLI YU YINGYONG

肖广润　周惠领　张鄂亮

华中理工大学出版社

高等学校教材

微型计算机原理与应用

8051、8098单片机应用基础

肖广润　周惠领　张鄂亮

华中理工大学出版社

（鄂）新登字第 10 号

图书在版编目（CIP）数据
微型计算机原理与应用/肖广润 等
武汉：华中理工大学出版社，1996 年 9 月
ISBN 7-5609-1325-3
Ⅰ. 微…
Ⅱ. ①肖… ②周… ③张…
Ⅲ. 微型计算机-基本知识
Ⅳ. TP36

高等学校教材
微型计算机原理与应用
8051、8098 单片机应用基础
肖广润　周惠领　张鄂亮
责任编辑：钟小珉
*
华中理工大学出版社出版发行
（武昌喻家山　邮编：430074）
新华书店湖北发行所经销
武汉大学印刷厂印刷
*
开本：787×1092　1/16　印张：22　字数：528 000
1996 年 9 月第 1 版　1999 年 4 月第 3 次印刷
印数：7 001-9 500
ISBN　7-5609-1325-3/TP・181
定价：22.00
（本书若有印装质量问题，请向出版社发行部调换）

内 容 提 要

本书从微型计算机的原理入手，比较全面、系统地介绍了 8051 和 8098 单片机的硬件、软件及应用技术。内容包括微型计算机基础知识，单片机结构原理，指令系统与程序设计，接口电路与单片机系统扩展，以及接口应用技术等。全书共分八章，每章配有习题。

本书内容由浅入深、循序渐进，适合作为高等院校有关专业学生的教材和工程技术人员的参考书。

前　言

为了适应微型计算机技术飞速发展的需要，目前理工科高等院校普遍开设了《微型计算机原理与应用》课程。近十年来，单片机以其体积小、价格低、功能齐全、抗干扰性高、可靠性好、易于开发扩展等独特优点，表现出强大的生命力。它广泛应用于工业自动化及智能仪器仪表、智能设备、家用电器等各种设备、仪器中，并已渗透到生产、经济、生活等各个领域中。

目前由美国 Intel 公司生产的 MCS-51、MCS-96 系列是我国单片机应用的主流机种。本书着重介绍 8051 和 8098 这两种最常用的机型。考虑到理工科院校的学生和初步接触微型计算机的工程技术人员学习单片机的需要，本书各部分内容都从微机原理的基础知识入手，采用由浅入深、从一般到特殊的方式，逐步向单片机过渡。这对初学者和有一定微机基础的读者，掌握单片机的应用技术将会有所帮助。

本书的另一个特点是：它不是孤立地分开介绍 8051 和 8098 这两种机型，而是尽可能采用并列对比的方式进行叙述。在着重介绍不同机型各自特点的同时，也展示了它们在应用中的兼容性。读者可以重点掌握一种机型，同时了解另一种机型，也可以两种机型一起学习。我们认为，这种安排不仅节省篇幅，而且还能在一定程度上促进对单片机知识的融会贯通。

本书例题丰富，解释详明，并具有实用性，对于读者弄清概念、掌握方法颇有裨益。而且，书中部分例题可抽出直接用于实验教学。

本书可作为高等院校“微型计算机原理与应用”课程的教材，并对从事微机应用的工程技术人员具有参考价值。

参加本书编写的还有广东工业大学谭向红老师。

由于作者水平有限，编写时间仓促，书中错漏在所难免，敬请读者批评指正。

作者

1995 年 12 月

目　　录

第一章　计算机基础知识

1-1　计算机的发展与组成

这一节通过简要介绍计算机的发展与组成结构，使读者了解计算机的有关知识。

1-1-1　计算机发展概况

自从1946年第一台数字电子计算机诞生至今，计算机已经历了由第一代到第四代的迅猛发展历史。第一代计算机的逻辑元件为电子管。第二代是于1958年制成的以晶体管为逻辑元件的计算机。由于集成电路的出现，1964年制成集成电路的计算机，为第三代计算机。1970年出现超大规模集成电路的计算机，即第四代计算机。现在正在研制和发展的人工智能计算机为第五代计算机。

微型计算机是超大规模集成电路孕育的产物。因为它具有体积小、价格便宜的优点，所以使计算机技术渗透到各行各业，甚至进入千家万户；反过来，其广泛的应用前景也大大促进了计算机技术高速地发展。

单片机是一种集成度更高的微型计算机，在一块小芯片上就集成了一台计算机。单片机具有高可靠性、高抗干扰能力、高性能价格比的优势，特别适用于实时工业测量控制、智能化仪器仪表、计算机外设、家用电器控制等应用系统。单片机应用范围如表1-1-1所示。

表1-1-1　单片机的应用范围

工业方面	汽车方面	导航与控制方面
电机控制	点火控制	导弹控制
工业机器人	变速器控制	鱼雷制导导弹控制
过程控制	防滑刹车	智能武器装置
数字控制	排气控制	航天导航系统
智能传感器	避雷控制	电子干扰装置
民用方面	节能控制	**数据处理方面**
电子玩具、字典、记事簿	保安控制	图表终端、图文传真机
高级电子游戏机	**仪器仪表方面**	彩色与黑白复印机
录像机	智能仪器	温氏硬盘驱动器
激光驱动、红外线驱动	医疗器械	磁带机、打印机、打字机
照相机、空调机、防盗控制	色谱仪	**电讯方面**
调制解调器	示波器	智能线路运行控制

自1976年Intel公司率先推出MCS-48系列8位单片机后，各种类型和型号的单片机犹如雨后春笋般相继问世。

1980年Intel公司推出的MCS-51系列高档8位单片机，功能丰富、使用灵活、“性能价格比”远高于MCS-48系列单片机。引进后受到工程技术人员的欢迎，成为我国8位机的主流机型。

目前MCS-51系列单片机的应用技术日趋成熟，适应各种应用场合的新品种不断出现，如

高速型、低功耗型、A/D 型、PWM 型、高级语言型等等。与 MCS-51 单片机配套的芯片、器件和开发装置的种类也很多，使用十分方便。

随着单片机应用范围的扩大，人们对单片机的性能要求越来越高，1983 年 Intel 公司又推出 MCS-96 系列高性能的 16 位单片机，它的处理速度更高，功能更齐备。但由于其价格昂贵，使这种单片机在我国的应用受到限制。

1988 年 Intel 研制出了 8098 准 16 位单片机，它属于 MCS-96 系列的子系列，其性能与 MCS-96 系列的 16 位机基本相同，而价格却与 MCS-51 系列 8 位机接近，因而也受到人们的青睐。

1-1-2 计算机发展趋势

当今计算机主要朝以下几个方向发展：

微型化 超大规模集成电路产品几乎是两年一换代，使得计算机集成度越来越高，体积越来越小，而信息容量和运算速度以及性价比却越来越高。

巨型化 用于军事科学、空间技术和原子能等尖端科技领域的巨型计算机，要求信息容量特大，运算速度特快，功能极强。

智能化 为人工智能型计算机，可以模拟人的思维、推理和判断能力，能识别图形、声音、文字等各种自然语言。

系列化 为了便于计算机升级换代，减少用户的硬件、软件投资，计算机制造商采用兼容设计方法，使产品系列化。

组成计算机网络 应用计算机网络技术，可使各地计算机联网，通过通信线路交换信息，实现硬件、软件资源共享。

多机系统 多台微机组成主从式、分散综合式等多微机系统，完成一些大型复杂的数据采集和控制任务。系统内部体现独立控制、集中管理、资源共享、故障分散的原则，使生产、控制、管理科学化、规范化。

1-1-3 计算机的组成结构

一台计算机系统包含有硬件系统和软件系统两大部分，其组成结构如图 1-1-1 所示。计算机硬件系统的核心部分是运算器和控制器，两者合称为中央处理单元 CPU (Centrol Processing Unit)。CPU 常被比喻为计算机的大脑和心脏。

- 计算机系统
  - 硬件
    - 主机
      - 运算器
      - 控制器
      - 主存储器
      - 输入、输出接口部件
    - 外围
      - 外部设备
        - 外存储器
        - 输入设备
        - 输出设备
      - 辅助设备
        - 电源电路
        - 时钟电路
  - 软件
    - 系统软件(操作系统、编辑、编译程序，监控程序等)
    - 应用软件
    - 程序设计语言(机器语言、汇编语言、高级语言)

图 1-1-1 计算机的组成结构

利用大规模集成电路技术将运算器、控制器做在一块半导体芯片上，称作微处理器(Microprocessor)。微处理器加上大容量半导体存储器和各种功能的输入/输出接口电路的集成芯片，并用总线结构连为一个整体，就构成微型计算机(Microcomputer)。微型计算机主机配以必要的外部设备、电源辅助设备和软件系统，便组成一个微型计算机系统，如图 1-1-2 所示。

- 微型计算机系统
  - 硬件
    - 微型计算机
      - 微处理器(CPU)
        - 运算器
        - 控制器
      - 存储器(M)
        - RAM
        - ROM
      - 输入/输出接口(PIO、SIO、CTC…)(I/O 接口)
      - 总线(AB、DB、CB)
    - 外围设备
      - 外部设备
        - 外存储器(磁盘驱动器、磁带机、光盘机…)
        - 输入设备(键盘、光电扫描仪、语音识别仪…)
        - 输出设备(显示器、打印机、绘图仪、电传打字机…)
      - 辅助设备
  - 软件

图 1-1-2　微型计算机的组成结构

如果将 CPU、存储器和输入/输出接口集成在一块半导体芯片上，这便是单片机(Single-chip Microcomputer)。

输入/输出设备是计算机与外界联系的桥梁。人们通过输入设备向计算机输入程序和发送各种运行命令，通过输出设备了解计算机的运行状态和计算结果。输入/输出设备是计算机系统的一个重要组成部分。

能在计算机上运行的各种程序称为软件，主要有两大类：

系统软件　由计算机设计者提供的，为方便人们使用和管理计算机的软件，都称为系统软件。

应用软件　用户利用计算机的硬、软件系统为解决各种实际问题而编制的程序，称为应用软件。

设计程序时，可使用各种计算机语言。程序语言主要有三种：机器语言、汇编语言和高级语言。

1-1-4　计算机的主要技术指标

字长　计算机一次并行处理的二进制位数，称为字长。它影响着计算速度和精度。我们通常所说的 8 位机、16 位机、32 位机等，皆是以字长分类的。

主存储量　指计算机主存储器的容量。它影响着计算机的数据处理能力和速度以及软件配置。

运算速度　指计算机每秒钟所能执行的指令条数。根据不同类型的指令所需的执行时间的不同，有以下几种计算运算速度的方法：

1)每秒钟执行加法指令的条数；

2)计算机允许的最大主频(时钟频率)；

3)每条指令执行所需的机器周期。

主存储器的存取周期　指主存储器进行一次读写操作所需的时间。

此外还有最大寻址空间、I/O 接口的功能、可靠性能、抗干扰性能等项指标。

1-2 计算机运算基础

在计算机中，数字是用一串“0”或“1”的二进制代码来表示的。对于二进制来说，有一整套独特的运算规则。本节将首先从常用的十进制数开始分析，然后再列入各种不同进位计数制，并介绍它们之间的相互转换关系。

本节还将介绍计算机中常用的二进制编码，以及它们的运算方法。

1-2-1 进位计数制

日常生活中，最常用的进位计数制是十进制，而计算机知识中还要用到二进制和十六进制。

（一）十进制(Decimal)

十进制的计数特点是：①有 0，1，…，9 十个数码；②“逢十进一”。对位置记数制来说，一个数的大小决定于数码的大小以及数码所处的位置(即数位)。例如，一个十进制数 1995 可写成展开式：

$$1995=1\times10^3+9\times10^2+9\times10^1+5\times10^0$$

其中，10 称为基数，10^0，10^1，10^2，…称为各数位的“权”。十进制个位的权为 1，十位的权为 10，百位的权为 100，……。任何一个十进制数 N_D，可表示为：

$$N_D=d_{n-1}\times10^{n-1}+d_{n-2}\times10^{n-2}+\cdots+d_1\times10^1+d_0\times10^0+d_{-1}\times10^{-1}+\cdots+d_{-m}\times10^{-m}$$

$$=\sum_{i=-m}^{n-1}d_i\times10^i$$

式中，d_i 为各数位的系数，它为 0～9 十个数码之一，所对应的数值为 $d_i\times10^i$。

（二）二进制(Binary)

如果电子计算机使用十进制数，每位数的十个数码需要电路中的 10 个状态与之对应，这在实际中很难实现。而仅用电路的两个稳定状态就能很容易地表示每位数的两个数码，所以计算机内都使用二进制。

二进制只有 0、1 两个数码，且“逢二进一”。二进制以 2 为基数，它的权即为 2 的 n 次幂。一个二进制数 N_B 可写成：

$$N_B=b_{n-1}\times2^{n-1}+b_{n-2}\times2^{n-2}+\cdots+b_1\times2^1+b_0\times2^0-b_{-1}\times2^{-1}+\cdots+b_{-m}\times2^{-m}$$

$$=\sum_{i=-m}^{n-1}b_i\times2^i$$

例如，一个二进制数 1101.101 可展开为：

$$1101.101=1\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}$$

（三）十六进制(Hexadecimal)

二进制数的位数太多，不易书写和阅读，所以经常使用十六进制数来表示计算机内的二进制数码。

十六进制有 16 个数码：0，1，…，9，A，B，C，D，E，F，且“逢十六进一”。一个十六进制数 N_H 可表示为：

$$N_H = h_{n-1} \times 16^{n-1} + h_{n-2} \times 16^{n-2} + \cdots + h_0 \times 16^0 + h_{-1} \times 16^{-1} + \cdots + h_{-m} \times 16^{-m}$$
$$= \sum_{i=-m}^{n-1} h_i \times 16^i$$

h_i 为 0～F 的 16 个数码之一。十六进制以 16 为基数，第 i 位的权为 16^i。例如：

$$DFC.8 = D \times 16^2 + F \times 16^1 + C \times 16^0 + 8 \times 16^{-1}$$
$$= 13 \times 16^2 + 15 \times 16^1 + 12 \times 16^0 + 8 \times 16^{-1}$$

因使用多种进位计数制，通常用下标予以区别。用 D、B、H 分别表示十、二、十六进制数，例如：

$$(1995)_D = (7CB)_H = (11111001011)_B$$

有时不写下标，即表示为十进制数。如 $101 = (101)_D$。

任何进制数都可写成如下形式：

$$N = a_{n-1} \times r^{n-1} + a_{n-2} \times r^{n-2} + \cdots + a_0 \times r^0 + a_{-1} \times r^{-1} + \cdots + a_{-m} \times r^{-m}$$
$$= \sum_{i=-m}^{n-1} a_i \times r^i \qquad (1\text{-}2\text{-}1)$$

式中，r 为基数，a_i 为 0，1，…，r－1 中任一个数码，n 为整数位数，m 为小数位数。

上式是不同进位制数之间转换的基础。一般来说，如果要将一个 r_1 进制的数转换成 r_2 进制表示的数，先按(1-2-1)式展开，再按 r_2 进制的运算法则进行计算即可完成。

1-2-2　不同进位计数制之间的转换

为了能在不同应用场合用不同计数制来表示一个数，就需要掌握不同进制的转换关系。

（一）二、十六进制数转换成十进制数

我们十分熟悉十进制的运算法则，只要按(1-2-1)式展开后运算即可。例如：

$$(1011.1010)_B = 1\times2^3+1\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-3} = (11.625)_D$$
$$(DFC.8)_H = 13\times16^2+15\times16^1+12\times16^0+8\times16^{-1} = (3580.5)_D$$

（二）二进制与十六进制数之间的转换

因为 $2^4=16$，正好四位二进制数与一位十六进制数相对应。表 1-2-1 为二、十、十六进制数的对照表，熟记之可方便地进行不同数制之间的转换。例如：

表 1-2-1　不同进位计数制对照表

十进制	二进制	十六进制	十进制	二进制	十六进制
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

$$(3AF.2)_H = \underbrace{0011}_{3}\,\underbrace{1010}_{A}\,\underbrace{1111}_{F}\,.\,\underbrace{0010}_{2} = (111010\ 1111.001)_B$$

$$(1101011.11)_B = \underbrace{0110}_{6}\underbrace{1011}_{B}.\underbrace{1100}_{C} = (6B.C)_H$$

以小数点为界，整数自右向左每四位一分，不足前面补 0；小数从左往右每四位一分，不足后面补 0。对照表 1-2-1 写出十六进制数。

可见，二进制与十六进制数之间的转换简捷方便，这便是十六进制的用途所在。计算机中二进制数码的基本位数都是 4 的整数倍，如 8 位、16 位、32 位等。所以很容易用十六进制数来表示。

（三）十进制数转换成二、十六进制数

1. 整数转换法

整数转换一般采用“除基取余”法。将十进制整数不断除以转换进制的基数，并取其余数，直至商为 0。第一次所得余数为转换进制最低整数位系数。下面举例说明。

[例 1-2-1] 将十进制整数 39 转换成二进制数。

解 设 39 所对应的二进制数表示如下：

$$(39)_D = b_{n-1} \times 2^{n-1} + \cdots + b_1 \times 2^1 + b_0 \times 2^0$$

如果将两边同除以二进制的基数 2，由上式可见，39/2＝整数＋余数/2，所得余数必然等于 b_0。再依次除 2 取余，便可得到转换后二进制各数位的系数 $b_1, b_2 \cdots$。转换过程如下：

$$\begin{array}{r|l}
2 & 39 \quad \cdots \quad 余\ 1\ (b_0) \\ \hline
2 & 19 \quad \cdots \quad 余\ 1\ (b_1) \\ \hline
2 & 9 \quad \cdots \quad 余\ 1\ (b_2) \\ \hline
2 & 4 \quad \cdots \quad 余\ 0\ (b_3) \\ \hline
2 & 2 \quad \cdots \quad 余\ 0\ (b_4) \\ \hline
2 & 1 \quad \cdots \quad 余\ 1\ (b_5) \\ \hline
 & 0
\end{array}$$

可得转换结果：

$$(39)_D = (100111)_B$$

验证如下：

$$(100111)_B = 1 \times 2^5 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 32 + 4 + 2 + 1 = (39)_D$$

[例 1-2-2] 将十进制整数 208 转换成十六进制数。

解 十六进制的基数为 16，除基所得余数可为 0～F 中任一数码。转换过程如下：

$$\begin{array}{r|l}
16 & 208 \quad \cdots \quad 余\ 0 \\ \hline
16 & 13 \quad \cdots \quad 余\ 13\ 即(D)_H \\ \hline
 & 0
\end{array}$$

所以，

$$(208)_D = (D0)_H$$

[例 1-2-3] 将十进制整数 123456 转换成二进制数。

解 先转换成十六进制，再直接写出二进制，比多次除 2 取余法快。

$$(123456)_D = (1E240)_H = (1\ 1110\ 0010\ 0100\ 0000)_B$$

2. 小数转换法

小数转换可用“乘基取整”法。第一次乘基所得整数为转换进制的最高小数位系数。

［例 1-2-4］ 将十进制小数 0.625 转换成二进制数。

解 小数转换后也为小数，可写成以下转换式：

$$(0.625)_D=b_{-1}\times 2^{-1}+b_{-2}\times 2^{-2}+\cdots+b_{-m}\times 2^{-m}$$

当上式两边同乘以 2 后，两边整数部分与小数部分应分别相等。则 0.625 乘 2 所得整数必为 b_{-1}，将余下小数部分继续乘 2，可得 b_{-2}，b_{-3}，…。转换过程如下：

$2\times 0.625=1.25$ 取整为 1 (b_{-1})

$2\times 0.25=0.5$ 取整为 0 (b_{-2})

$2\times 0.5=1.0$ 取整为 1 (b_{-3})，小数为 0 结束

转换结果为：

$$(0.625)_D=(0.101)_B$$

转换过程有可能小数部分永不为零，可根据精度要求决定转换后的小数位数。

［例 1-2-5］ 将十进制小数 0.625 转换成十六进制数。

解 $16\times 0.625=10.0$ 取整为 $(A)_H$

∴ $(0.625)_D=(0.A)_H$

［例 1-2-6］ 将十进制数 208.625 转换成二、十六进制数。

解 将整数部分与小数部分分别转换得：

$$(208.625)_D=(D0.A)_H=(1101\ 0000.101)_B$$

1-2-3 带符号数的表示方法

（一）机器数与真值

在计算机中，不仅所有的数都是用“0”和“1”两个数码的组合来表示，数的符号也只能用“0”和“1”两个数码来表示。通常，将一个有符号数的最高位用作符号位，以“0”表示“+”号，以“1”表示“−”号。这就是说，数的符号在计算机中也数码化了。我们把一个数在机器中的表示形式称为机器数，而把机器数所代表的实际数值称为该机器数的真值。例如，一个 8 位机器数与它的真值关系可表示如下：

机器数：$x_1=01010100$　　　　真值：$x_1=(+1010100)_B=+84$

$x_2=11010100$　　　　$x_2=(-1010100)_B=-84$

在微机中，有符号数常用的表示法有两种：原码和补码。

（二）原码(True Form)

原码表示法约定，符号位为“0”，表示正数；符号位为 1，表示负数。其余位与实际数值位相同。

如上例所述，即为 8 位原码表示。通常写成如下形式：

$x_1=+84$　　$[x_1]_{原}=01010100$

$x_2=-84$　　$[x_2]_{原}=11010100$

8 位原码最高位是符号位，其余 7 位为数值位，因此所能表示的范围为$(-1111111)_B$～$(+1111111)_B$，即−127～+127。如果用 16 位原码表示，最高位仍作符号位，余下有 15 个数值位，所能表示的范围为−32767～+32767。

原码表示简单直观，可直接看出数值大小，但它有两个缺点。

1)零的表示不唯一。如 8 位原码有如下两种表示：

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

2)作加减运算很复杂。计算机中如果用原码作加减运算，先必须检查参加运算的两个数的符号，以决定实际应作加法还是作减法。如果是作减法，还要先比较两个数的绝对值大小，用大数减小数才能保证正确结果。最后还要判断结果的符号。

（三）补码(Two's Complement)

补码表示法克服了原码的上述两个缺点，能使有符号数的加减运算非常方便。这使计算机内的运算电路及运算过程大为简化。

补码表示法约定：正数的补码表示与原码相同；负数的补码符号位为 1，而数值位为原码数值各位取反，且在最低位上加 1。

例如，前述中 $x_1=+84$，$x_2=-84$，用 8 位补码表示为：

$$[x_1]_{补}=[x_1]_{原}=01010100,\ [x_2]_{补}=10101011+1=10101100$$

零的补码表示是唯一的：$[+0]_{补}=[-0]_{补}=00000000$，数值为0的最小负数10000000定为－128。因此 8 位补码的表示范围为－128～＋127。

16 位补码的表示范围为－32768～＋32767。例如，$x_1=+84$，$x_2=-84$，用 16 位补码表示为：

$$[x_1]_{补}=0000000001010100$$

$$[x_2]_{补}=1111111110101100$$

补码表示的主要优点是加减运算方便，可将有符号数的减法运算转为加法。下一节将作详细介绍。

日常生活中有不少补码的应用。如钟表对时。假设标准时间是 7 点整，而一只钟表指针指向 9 点整，这时钟表对时可有两种方式。

1)倒拨 2 小时：9－2＝7(点)

2)顺拨 10 小时：9＋10＝12＋7＝7(点)

由于钟表的 12 点与 0 点重合，12 被自然丢弃。所以(9－2)与(9＋10)是等价的，等价的条件是以 12 为模。所谓模是一个系统的量程，即此系统所能表示的最大数。对模为 12 的系统来说，2 与 10 互为补数(补码)，减 2 的运算可用加 10 的运算代替。此外，3 与 9、4 与 8、6 与 6 等也都在模 12 的系统中互为补数。

再如一个两位十进制的里程表，量程为 100km，即 100km 与 0km 的指示是重合的。设里程表初始位置为 0，前进 1km，表指示为 01，而后退 1km，表的指示为 99。因此对于模为 100 的两位十进制系统来说，减 1(后退 1km)运算相当于加 99。这里 1 与 99 互为补数，同样 2 与 98、3 与 97…也互为补数。

上面两个例子中，计算的方式为：

$$x_1-x_2=x_1+(模-x_2)=x_1+(-x_2)_{补}$$

因此，一个负数的补数，是用模减去该数的绝对值来求得。如前例中：

$$(-2)_{补}=12-2=10 \quad (模为 12)$$

$$(-1)_{补}=100-1=99 \quad (模为 100)$$

对于一个 8 位二进制数，模为 $2^8=256$，同样，一个负数 x 的补码(二进制码的补数常称为补码)可用下式求得：

$$[x]_{补}=256-|x|$$

问题是这种求法仍未避免减法，但根据二进制的特点，$256-|x|=255-|x|+1$，而 $255-|x|$ 的运算结果与 x 数值各位取反的结果完全一样。我们知道计算机电路中对二进制各位取反是件轻而易举的事。求带符号位的 8 位二进制负数 x 的补码，算式 $255-|x|+1$ 正好是将 x 绝对值的原码带符号一起按位取反。例如，求 $x=-84$ 的补码(模为 2^8)。

$$x=-84=-1010100_B, |x|=+1010100$$

$$[|x|]_{原}=[|x|]_{补}=01010100$$

$$[x]_{补}=255-[|x|]_{原}+1=11111111-01010100+1$$

$$=10101011+1=10101100$$

(四)机器数与真值之间的转换

我们看到，一个有符号数，由于编码方法不同，可能有几种机器数。反之，当约定的表示方法不同，同一个机器数，也可代表几种真值。表 1-2-2 列出了 8 位机器数在不同表示法下所代表的真值。因此，进行真值与机器数之间的转换，首先要明确其表示方法。下面仅讨论有符号数真值与原码、补码之间的转换。

表 1-2-2 8 位机器数的真值表

机器数：十六进制表示	机器数：二进制代码	十进制表示的真值（不同表示法）：无符号二进制数	有符号二进制数：原码	有符号二进制数：补码
00	00000000	0	+0	+0
01	00000001	1	+1	+1
02	00000010	2	+2	+2
⋮	⋮	⋮	⋮	⋮
7E	01111110	126	+126	+126
7F	01111111	127	+127	+127
80	10000000	128	−0	−128
81	10000001	129	−1	−127
⋮	⋮	⋮	⋮	⋮
FE	11111110	254	−126	−2
FF	11111111	255	−127	−1

1. 已知真值求原码和补码

求原码的方法是将真值符号位用二进制码表示，数值位保持不变。

求补码的第一步先求出原码，正数补码等于其原码；负数补码是将原码数值各位取反并在最低位上加 1，符号位与原码相同。

[**例 1-2-7**] 已知 $x_1=+127, x_2=-127$，求原码和补码。

解 $x_1=(+1111111)_B, x_2=(-1111111)_B$，可用 8 位二进制数码表示：

$$[x_1]_{原}=01111111,\ [x_2]_{原}=11111111$$

$$[x_1]_{补}=01111111,\ [x_2]_{补}=10000000+1=10000001$$

[**例 1-2-8**] 已知 $x_1=+255, x_2=-255$，求其原码和补码。

解 $x_1=+11111111_B,\ x_2=-11111111_B$，数值位有 8 位，还要加一位符号位，因此无法用

8 位二进制码表示。计算机的字长一般都是 8 的倍数，当 8 位不够，可改用 16 位二进制码表示，但要注意，求负数补码时，模应为 2^{16}。

$$[x_1]_原=0000000011111111, [x_2]_原=1000000011111111$$

$$[x_1]_补=0000000011111111, [x_2]_补=1111111100000001$$

2. 已知原码、补码求真值

[例 1-2-9] 已知$[x_1]_原=01011001$，$[x_2]_原=11011001$，求真值。

解 只需将原码符号位上的"0"变为"+"，"1"变为"−"即可求得。

$$x_1=+1011001_B=+89, x_2=-1011001_B=-89$$

[例 1-2-10] 已知$[x_1]_补=01011001$，$[x_2]_补=11011001$，求真值。

解 要注意正数和负数的补码表示是不同的。转换前要检查符号位，符号位为"0"，则为正数补码。真值等于"+"号后数值位照写；若符号位为"1"，则为负数补码。将补码数值各位取反再加 1（补数互补），前面加"−"号便还原成其二进制真值。

$$x_1=+1011001_B=+89, x_2=-0100111_B=-39$$

1-2-4 数的定点与浮点表示

在计算机中，数有定点和浮点两种表示法。小数点位置固定不变的数称为定点数。小数点位置可以浮动的数称为浮点数。

（一）定点数表示

定点数有以下两种表示法。

定点整数 规定小数点的位置固定在数的最低位之后。如表 1-2-2 中用 8 位二进制代码表示无符号和有符号数，实际都属于定点整数表示。定点整数可以表示无符号和有符号纯整数。

定点小数 规定小数点的位置固定在数的最高位之前。如果表示一个有符号数，小数点在符号位之后，数值位之前。定点小数可用来表示无符号和有符号纯小数。

设有两个有符号二进制数 +1011010 和 +0.1011010，如果分别用定点整数和定点小数表示，结果都为 01011010。可见，在计算机中，小数点并不占数码位，而是根据事先约定的表示法来得知数的大小和小数点的位置。

当一个数既有整数部分又有小数部分时，通常将整数部分用定点整数表示，小数部分用定点小数表示，并存放在计算机的两个连续数据单元中。

定点数的优点是运算简便且运算速度快，符合一般简单工业测控系统的要求。本书重点介绍定点数。

定点数的缺点是数的表示范围太小（如双字节有符号整数的定点表示范围为 −32768～+32767）。当数的范围很大或精度要求很高时，可采用浮点数表示。

（二）浮点数表示

一个二进制数可写成如下形式：

$$B=\pm S\times 2^{\pm J}$$

S 称为尾数，J 称为阶码。定点数表示中，J 恒为"0"，S 可为整数或小数。浮点数表示中，S 可为整数或小数，J 为整数。计算机中的浮点数包含阶码和尾数两部分。如果用两个数据单元存放一个浮点数，一般为如下形式：