

21世纪高等院校计算机教材系列

单片机原理 及应用教程

第 2 版

● 赵全利 肖兴达 主编

购书可获得增值回报
提供教学用电子教案

 机械工业出版社
CHINA MACHINE PRESS



21 世纪高等院校计算机教材系列

单片机原理及应用教程

第 2 版

赵全利 肖兴达 主编



机械工业出版社

本书详尽地介绍了 MCS-51 系列单片机的硬件结构、指令系统及汇编语言程序设计、内部功能及接口、系统外部功能的扩展、C51 程序设计、单片机系统开发及应用实例。书中通过大量的例题和单片机应用实例，引导读者逐步掌握单片机应用系统开发的基本知识、方法和技能。各章后配有习题，以巩固学生所学的知识。书中所有程序均经过了上机调试。

本书可作为高等院校电气信息类专业的教学用书，也可作为高职高专同类专业的教学用书及各类工程技术人员的自学用书。

图书在版编目 (CIP) 数据

单片机原理及应用教程/赵全利, 肖兴达主编. —2 版. —北京: 机械工业出版社, 2007.7

(21 世纪高等院校计算机教材系列)

ISBN 978-7-111-12530-3

I. 单… II. ①赵…②肖… III. 单片微型计算机—高等学校—教材
IV. TP368. 1

中国版本图书馆 CIP 数据核字 (2007) 第 098579 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 赵 慧

责任印制: 杨 曦

三河市国英印务有限公司印刷

2007 年 7 月第 2 版·第 1 次印刷

184mm×260mm·16.5 印张·406 千字

21001—26000 册

标准书号: ISBN 978-7-111-12530-3

定价: 25.00 元

凡购本书, 如有缺页, 倒页, 脱页, 由本社发行部调换

销售服务热线电话: (010) 68326294

购书热线电话: (010) 88379639 88379641 88379643

编辑热线电话: (010) 88379739

封面无防伪标均为盗版

出版说明

信息技术是当今世界发展最快、渗透性最强、应用最广的关键技术，是推动经济增长和知识传播的重要引擎。在我国，随着国家信息化发展战略的贯彻实施，信息化建设已进入了全方位、多层次推进应用的新阶段。现在，掌握计算机技术已成为 21 世纪人才应具备的基本素质之一。

为了进一步推动计算机技术的发展，满足计算机学科教育的需求，机械工业出版社聘请了全国多所高等院校的一线教师，进行了充分的调研和讨论，针对计算机相关课程的特点，总结教学中的实践经验，组织出版了这套“21 世纪高等院校计算机教材系列”。

本套教材具有以下特点：

- (1) 反映计算机技术领域的新发展和新应用。
- (2) 注重立体化教材的建设，多数教材配有电子教案、习题与上机指导或多媒体光盘等。
- (3) 针对多数学生的学习特点，采用通俗易懂的方法讲解知识，逻辑性强、层次分明、叙述准确而精炼、图文并茂，使学生可以快速掌握，学以致用。

(4) 符合高等院校各专业人才的培养目标及课程体系的设置，注重培养学生的应用能力，强调知识、能力与素质的综合训练。

(5) 适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班和自学用书。

机械工业出版社

2005 年 7 月

前 言

本书第 1 版自 2003 年 7 月出版以来, 被许多高等学校作为相关专业“单片机原理及应用”课程的教学参考书, 受到广大读者的欢迎, 在此我们向阅读、使用和关心此书的朋友们表示衷心的感谢。

随着计算机技术的高速发展和日益增强的社会需求, 单片机应用已广泛深入到各个领域, 单片机应用技术已成为电子技术产业的核心。为了适应新形势下学习单片机的需要, 本书作者从单片机应用开发的角度出发, 在本书第 1 版的基础上进行修改、调整和扩充, 以利于广大读者循序渐进地学习单片机, 更加适应各类高等学校、高职高专的教学需要。

事实已经证明, 以 8 位单片机组成的单片机应用系统, 以其通用性强、价廉、设计灵活等特点而遍及各个领域, 有着广泛的和稳定增长的市场。

本书以单片机应用为主要目的, 结合高等教育各专业的特点, 自成体系。首先介绍计算机基础知识及计算机系统组成, 然后, 以当前仍处于繁荣之势的 Intel 公司的 MCS-51 系列单片机为对象, 介绍其内部资源、程序设计及实例、硬件资源应用、系统扩展、人机界面、过程通道、Cx51 程序设计等应用开发中涉及到的问题。

本书概念清楚, 注重知识的内在联系与规律, 采用归纳、类比的方法, 系统介绍了单片机的结构原理及应用系统的组成与设计方法, 目的是使读者在较短的时间内掌握单片机的应用技术。为了便于读者理解、掌握本书的内容, 每章均配有大量的例题与习题。

本书实例取材于工程实际, 内容覆盖面广, 条理清晰, 具有很高的参考价值, 本书不仅对学习掌握单片机, 而且对掌握实际单片机系统开发技术也具有较好的帮助作用。本教材本着一切内容为应用服务的宗旨, 在介绍单片机基本知识、基本结构、基本组成的基础上, 通过大量由浅入深的单片机应用实例, 引导读者逐步认识、熟知、应用单片机, 为单片机的开发和应用打下坚实的基础。

本教材是依据中国计算机学会高等教育学会最新审定的编写大纲编写的, 所有例题和源程序都经上机调试通过, 授课为 60 课时。

“单片机原理与应用”是高等学校电气信息类专业的一门专业必修课。随着单片机控制领域的不断拓展, 机械、机电类专业也开设了此课程。本书可作为高等学校相关专业“单片机原理与应用”课程的教学用书, 也可作为高职高专有关专业的教材, 也可供从事单片机系统开发、应用的工程技术人员阅读、参考。

本书由赵全利、肖兴达主编。参与本书编写工作的还有翟红程、刘绍宁、郝心、张险峰、白晗东、张宏敏、闫巧枝、黄颖辉、宋海军、郭银、雷成茂、许镭、李聪、彭春艳。由于编者水平有限, 书中难免有错误与不妥之处, 请广大读者批评指正。

为了配合本书的教学, 机械工业出版社为读者提供了电子教案。可在机械工业出版社网站 (www.cmpbook.com) 下载。

作 者

2007 年 5 月

目 录

出版说明

前言

第 1 章 计算机基础知识概述	1
1.1 数制与编码	1
1.1.1 计算机与二进制	1
1.1.2 数制及其转换	1
1.1.3 编码	5
1.2 计算机系统组成	9
1.2.1 计算机硬件组成	9
1.2.2 计算机软件系统	11
1.2.3 计算机语言及程序设计	11
1.3 单片机与嵌入式系统组成	16
1.3.1 单片机及其发展概况	16
1.3.2 单片机的特点和应用	16
1.3.3 嵌入式系统	18
1.3.4 单片机应用系统的组成	19
1.4 一个简单的单片机应用实例	20
1.5 习题	21
第 2 章 MCS-51 单片机硬件结构	23
2.1 MCS-51 单片机总体结构	23
2.1.1 MCS-51 单片机总体结构框图及功能	23
2.1.2 MCS-51 引脚功能	27
2.2 MCS-51 存储结构及位处理器	30
2.2.1 MCS-51 存储器的特点	30
2.2.2 程序存储器	31
2.2.3 数据存储器	31
2.2.4 专用寄存器区 (SFR)	34
2.2.5 位处理器	35
2.3 MCS-51 工作方式和时序	36
2.3.1 复位方式	36
2.3.2 程序执行工作方式	37
2.3.3 节电工作方式	38
2.4 MCS-51 单片机的时序	39
2.4.1 时钟	39

2.4.2 CPU 时序	39
2.5 习题	42
第 3 章 MCS-51 指令系统及汇编语言程序设计	44
3.1 指令系统简介	44
3.1.1 指令格式	44
3.1.2 指令分类及符号说明	45
3.2 寻址方式	46
3.3 指令系统	50
3.3.1 数据传送类指令	50
3.3.2 算术运算类指令	57
3.3.3 逻辑运算类指令	60
3.3.4 控制转移类指令	63
3.3.5 位操作类指令	67
3.3.6 子程序调用与返回指令	70
3.4 汇编语言程序设计基础	72
3.4.1 汇编语言程序的组成	72
3.4.2 伪指令	73
3.4.3 程序设计步骤及技术	75
3.5 程序设计实例	76
3.5.1 汇编语言基本程序设计	76
3.5.2 延时程序设计	79
3.5.3 代码转换程序设计	80
3.5.4 查表程序设计	81
3.5.5 运算程序设计	81
3.5.6 排序程序设计	82
3.5.7 输入输出程序设计	83
3.5.8 数字滤波程序设计	84
3.6 习题	84
第 4 章 MCS-51 单片机的中断系统	86
4.1 中断的概念	86
4.1.1 中断及中断源	86
4.1.2 中断嵌套	86
4.2 MCS-51 中断系统结构及中断控制	87
4.2.1 中断源和中断请求标志	87
4.2.2 中断允许控制	89
4.2.3 中断优先级控制	89
4.3 MCS-51 中断响应过程	90
4.3.1 中断响应	90

4.3.2	中断处理	91
4.3.3	中断返回	91
4.4	中断响应时间	91
4.5	中断响应后中断请求的撤除	92
4.6	中断系统应用实例	92
4.6.1	单步操作的中断实现	92
4.6.2	外部中断源扩展	93
4.7	中断程序举例	94
4.8	习题	98
第5章 MCS-51 单片机定时器/计数器		99
5.1	定时器/计数器概述	99
5.2	定时器/计数器的控制	100
5.2.1	定时器/计数器的工作模式寄存器 TMOD	100
5.2.2	定时器控制寄存器 TCON	101
5.3	定时器/计数器的工作模式	101
5.3.1	工作模式 0	101
5.3.2	工作模式 1	102
5.3.3	工作模式 2	102
5.3.4	工作模式 3	103
5.4	定时器/计数器应用实例	104
5.4.1	模式 0 的应用	104
5.4.2	模式 1 的应用	105
5.4.3	模式 2 的应用	105
5.4.4	模式 3 的应用	106
5.4.5	定时器/计数器综合应用	107
5.5	习题	109
第6章 单片机串行口及应用		111
6.1	串行通信的基本概念	111
6.1.1	异步通信和同步通信	111
6.1.2	串行通信的制式	113
6.1.3	波特率和发送/接收时钟	113
6.1.4	奇偶校验	114
6.2	MCS-51 单片机串行口	114
6.2.1	串行口结构	114
6.2.2	串行口控制	115
6.2.3	串行口的工作方式	116
6.2.4	波特率设置	118
6.3	串行口应用	120

6.3.1	串行口方式 0 的应用	120
6.3.2	串行口在其他方式下的应用	122
6.3.3	双机通信	125
6.3.4	多机通信	127
6.4	常用串行通信总线标准及接口电路	133
6.4.1	RS-232C 总线标准及接口电路	134
6.4.2	RS-422/485 总线标准及接口电路	135
6.5	习题	137
第 7 章	单片机的系统扩展	138
7.1	单片机系统扩展概述	138
7.2	程序存储器的扩展	139
7.2.1	常用的程序存储器	139
7.2.2	程序存储器的扩展	143
7.3	数据存储器的扩展	145
7.3.1	常用的数据存储器	145
7.3.2	数据存储器的扩展	147
7.4	I/O 端口的扩展	149
7.4.1	简单并行 I/O 口的扩展	149
7.4.2	8155 可编程多功能接口的扩展	151
7.5	单片机扩展系统外部地址空间的编址方法	158
7.5.1	线选法	158
7.5.2	译码法	159
7.6	习题	160
第 8 章	单片机系统的接口	162
8.1	键盘及接口电路	162
8.1.1	键盘的分类	162
8.1.2	键盘的工作原理	162
8.1.3	键盘结构及扫描子程序	163
8.1.4	键盘接口扩展设计	165
8.2	显示及显示器接口	167
8.2.1	LED 状态显示	167
8.2.2	LED 数码显示	169
8.2.3	七段 LED 数码管显示接口	171
8.2.4	键盘、显示器综合设计	172
8.3	A/D、D/A 转换器与单片机的接口	175
8.3.1	D/A 转换器	176
8.3.2	A/D 转换器	179
8.4	应用举例	184

8.4.1	简易 5V 直流数字电压表	184
8.5	习题	186
第 9 章	单片机 C 语言程序设计	188
9.1	Keil C 简介与环境设置	188
9.2	Cx51 简介	189
9.2.1	Cx51 的扩展	189
9.2.2	存储区	190
9.2.3	存储模式	190
9.2.4	数据类型	191
9.2.5	指针	193
9.2.6	函数	193
9.3	Cx51 开发工具简介	196
9.4	Cx51 应用程序设计举例	200
9.4.1	输入	200
9.4.2	输出	204
9.4.3	定时器/计数器	205
9.4.4	外部中断	207
9.4.5	串口通信	208
9.5	习题	210
第 10 章	单片机应用举例与系统开发	211
10.1	单片机应用系统举例	211
10.1.1	光电计数器	211
10.1.2	时间顺序控制器	213
10.1.3	转速表	217
10.1.4	交通灯管理系统	219
10.2	单片机系统的开发应用过程	224
10.2.1	总体论证	224
10.2.2	总体设计	226
10.2.3	硬件设计	226
10.2.4	软件设计	226
10.2.5	联机调试	227
10.2.6	脱机运行	228
10.3	单片机开发系统	228
10.3.1	开发系统的基本功能	228
10.3.2	开发系统简介	228
10.4	单片微机系统应用开发举例	231
10.4.1	系统简介	231
10.4.2	总体设计与模块功能分配	231

10.4.3	单元控制器硬件设计	233
10.4.4	单元控制器软件设计	234
10.5	习题	234
附录	236
附录 A	MCS-51 指令表	236
附录 B	ASCII (美国标准信息交换码) 码表	240
附录 C	典型汇编及模拟运行程序的使用	241
C.1	源程序汇编	241
C.2	源程序的模拟运行	246

第1章 计算机基础知识概述

本章主要讲述计算机的基础知识，包括：常用的数制及其不同数制相互间的转换、编码；计算机系统组成、接口、总线及程序设计的基本知识；单片机的概念、特点、应用领域及其嵌入式系统的概念。最后通过一个简单的单片机应用实例使读者初步了解单片机的工作概况。

1.1 数制与编码

在计算机中，任何命令和信息都是以二进制数据的形式存储的。计算机所执行的全部操作都归结为对数据的处理和加工，为了便于理解计算机系统的基本工作原理，掌握数字、字母等字符在计算机系统内的表示方法及处理过程，本节主要介绍有关计算机与数制和编码等方面的基础知识。

1.1.1 计算机与二进制

计算机科学技术的发展在不断地改变着世界，计算机所具有的神奇般的功能对人类社会所产生的重大影响，使一些人仍然把它看得十分神秘。其实，计算机只是一种以二进制数据形式存储信息、以程序存储为基础、由程序自动控制的电子设备。

在计算机中，由于所采用的电子逻辑器件仅能存储和识别两种状态的特点，计算机内部一切信息存储、处理和传送均采用二进制数的形式。可以说，二进制数是计算机硬件能直接识别并进行处理的惟一形式。

人们需要计算机所作的任何工作，都必须以计算机所能识别的指令形式转换为二进制数据送入计算机内存中，一条条有序指令的集合称为程序。计算机的工作过程也就是执行程序的过程，计算机所作的任何工作都是执行程序的结果。可以说，二进制数据存储信息和程序存储是计算机的基本工作机制。

1.1.2 数制及其转换

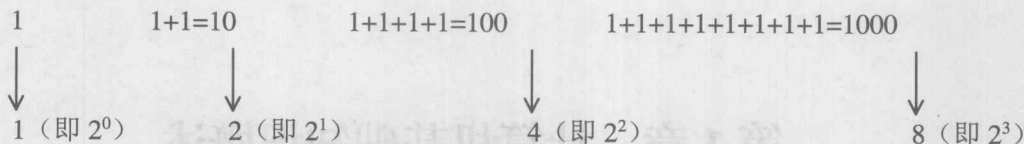
数制就是计数方式。

日常生活中常用的是十进制计数方式，计算机内部使用的是二进制数据，在向计算机输入数据及输出数据时，人们惯于用十进制、十六进制数据等，因此，计算机在处理数据时，必须进行各种数制之间的相互转换。

1. 二进制数

二进制数只有两个数字符号：0 和 1。计数时按“逢二进一”的原则进行计数。也称其基数为二。一般情况下，二进制数可表示为 $(110)_2$ 、 $(110.11)_2$ 、10110B 等。

根据位权表示法，每一位二进制数在其不同位置表示不同的值。例如：



对于 8 位二进制数（由低位~高位分别用 D0~D7 表示），则各位所对应的权值为：

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
D7	D6	D5	D4	D3	D2	D1	D0

对于任何二进制数，可按位权求和展开为与之相应的十进制数，则有：

$$(10)_2 = 1 \times 2^1 + 0 \times 2^0 = (2)_{10}$$

$$(11)_2 = 1 \times 2^1 + 1 \times 2^0 = (3)_{10}$$

$$(110)_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (6)_{10}$$

$$(111)_2 = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (7)_{10}$$

$$(1111)_2 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (15)_{10}$$

$$(10110)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (22)_{10}$$

例如，二进制数 10110111，按位权展开求和计算可得：

$$\begin{aligned} (10110111)_2 &= 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 128 + 0 + 32 + 16 + 0 + 4 + 2 + 1 \\ &= (183)_{10} \end{aligned}$$

对于含有小数的二进制数，小数点右边第一位小数开始向右各位的权值分别为：

2^{-1}	2^{-2}	2^{-3}	2^{-4}	...

例如，二进制数 10110.101，按位权展开求和计算可得：

$$\begin{aligned} (10110.101)_2 &= 1 \times 2^4 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} \\ &= 16 + 4 + 2 + 0.5 + 0.125 \\ &= (22.625)_{10} \end{aligned}$$

必须指出：在计算机中，一个二进制数（如 8 位、16 位或 32 位）既可以表示数值，也可以表示一种符号的代码，还可以表示某种操作（即指令），计算机在程序运行时按程序的规则自动识别，这就是本节开始所述及的，即一切信息都是以二进制数据进行存储的。

2. 十六进制数

十六进制数是学习和研究计算机中二进制数的一种比较方便的工具。计算机在信息输入输出或书写相应程序或数据时，可采用简短的十六进制数表示相应的位数较长的二进制数。

十六进制数有 16 个数字符号，其中 0~9 与十进制相同，剩余 6 个为 A~F，分别表示十进制数的 10~15，见表 1-1。十六进制数的计数原则是逢“十六进一”，也称其基数为十六，整数部分各位的权值由低位~到高位分别为： 16^0 、 16^1 、 16^2 、 16^3。例如：

$$(31)_{16} = 3 \times 16^1 + 1 \times 16^0 = (49)_{10}$$

$$(2AF)_{16} = 2 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 = (687)_{10}$$

为了便于区别不同进制的数据，一般情况下可在数据后面跟一后缀：

二进制数用“B”表示(如 00111010B);

十六进制数用“H”表示(如 3A5H);

十进制数用“D”表示(如 39D 或 39)。

3. 不同数制之间的转换

前已述及, 计算机中的数只能用二进制表示, 十六进制数适合读写方便的需要, 日常生活中使用的是十进制数, 计算机必须根据需要对各种进制数据进行转换。

(1) 二进制数转换为十进制数

对任意二进制数均可按权值展开将其转化为十进制数。例如:

$$10111B=1 \times 2^4+0 \times 2^3+1 \times 2^2+1 \times 2^1+1 \times 2^0=23D$$

$$10111.011B=1 \times 2^4+0 \times 2^3+1 \times 2^2+1 \times 2^1+1 \times 2^0+0 \times 2^{-1}+1 \times 2^{-2}+1 \times 2^{-3} \\ =23.375D$$

(2) 十进制数转换为二进制数

方法 1:

十进制数转换为二进制数, 可将整数部分和小数部分分别进行转换, 然后合并。其中整数部分可采用“除 2 取余法”进行转换。小数部分可采用“乘 2 取整法”进行转换。

例如: 采用“除 2 取余法”将 37D 转换为二进制数。

$\begin{array}{r} 2 \overline{)37} \\ 2 \overline{)18} \\ 2 \overline{)9} \\ 2 \overline{)4} \\ 2 \overline{)2} \\ 2 \overline{)1} \\ 0 \end{array}$	余数 1 0 1 0 0 0 1	低位 (第一次余数必为低位) ↓ 高位 (直到商数等于 0 为止)
--	---------------------------------------	---

把所得余数由高到低排列起来可得:

$$37=100101B$$

例如: 采用“乘 2 取整法”将 0.625 转换为二进制数小数。

$\begin{array}{r} 0.625 \\ \times 2 \\ \hline 1.250 \end{array}$	取整数 1	高位 (第一次整数 1 必为二进制数小数权值最高位)
$\begin{array}{r} 0.500 \\ \times 2 \\ \hline 1.000 \end{array}$	取整数 0	↓
$\begin{array}{r} 1.000 \\ \times 2 \\ \hline 2.000 \end{array}$	取整数 1	低位

把所得整数由高到低排列起来可得:

$$0.625=0.101B$$

同理, 把 37.625 转换为二进制数, 只须将以上转换合并起来可得:

$$37.625=100101.101B$$

方法 2:

可将十进制数与二进制位权从高位到低位比较, 若十进制数大于或等于二进制某位, 则该位取“1”, 否则该位取“0”, 采用按位分割法进行转换。

例如: 将 37.625 转换为二进制数。

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	0	1	0	0	1	0	1

将整数部分 37 与二进制各位权值从高位→低位比较, $37 > 32$, 则该位取 1, 剩余 $37 - 32 = 5$, 逐位比较, 得 00100101B。

将小数部分 0.625 按同样方法, 得 0.101B。

结果为 $37.625D = 100101.101B$ 。

(3) 二进制数与十六进制数的相互转换

在计算机进行输入、输出时, 常采用十六进制数。十六进制数可看作是二进制数的简化表示。

因为 $2^4 = 16$, 所以 4 位二进制数相当于 1 位十六进制数, 二进制、十进制、十六进制对应数的转换关系见表 1-1。

表 1-1 二进制、十进制、十六进制转换表

十 进 制	二 进 制	十 六 进 制
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

在将二进制数转换为十六进制数时, 其整数部分可由小数点开始向左每 4 位为一组进行分组, 直至高位。若高位不足 4 位, 则补 0 使其成为 4 位二进制数, 然后按表 1-1 对应关系进行转换。其小数部分由小数点向右每 4 位为一组进行分组, 不足 4 位则末位补 0 使其成为

4 位二进制数，然后按表 1-1 对应关系进行转换。

例如：1000101B=0100 0101B=45H

10001010B=1000 1010B=8AH

100101.101B=0010 0101.1010B=25.AH

需要将十六进制数转换为二进制数时，则为上述方法的逆过程。

例如：45.AH=0100 0101.1010 B

例如：7ABFH=0111 1010 1011 1111 B

7 A B F

即 7ABFH =1111010101111111B

(4) 十进制数与十六进制数的相互转换

十进制数与十六进制数的相互转换可直接进行，也可先转换为二进制数，然后再把二进制数转换为十六进制数或十进制数。

例如：将十进制数 37D 转为十六进制数。

37D=100101B=00100101B=25H

例如：将十六进制数 41H 转换为十进制数。

41H=01000001B=65D

也可按位权展开求和方式将十六进制数直接转换为十进制数，这里不再详述。

1.1.3 编码

计算机通过输入设备（如键盘）输入信息和通过输出设备输出信息是多种形式的，既有数字（数值型数据）、也有字符、字母，各种控制符号及汉字（非数值型数据）等。计算机内部所有数据均用二进制代码的形式表示，前面所提到的二进制数，没有涉及到正、负符号问题，实际上是一种无符号数的表示，在解决实际问题中，有些数据确有正、负之分。为此，需要对常用的数据及符号等进行编码，以表示不同形式的信息。这种以编码形式所表示的信息既便于存储，也便于由输入设备输入信息、输出设备输出相应的信息。

1. 二进制数的编码

(1) 机器数与真值

一个数在计算机中的表示形式叫做机器数，而这个数本身（含符号“+”或“-”）称为机器数的真值。

通常在机器数中，用最高位“1”表示负数，“0”表示正数（以下均以 8 位二进制数为例）。

例如，设两个数为 N1、N2，其真值为：

N1=105=+01101001B

N2=-105=-01101001B

则对应的机器数为：

N1=0 1101001B（最高位“0”表示正数）

N2=1 1101001B（最高位“1”表示负数）

必须指出，对于一个有符号数，可因其编码不同而有不同的机器数表示法，如下面将要

介绍的原码、反码和补码。

(2) 原码、反码和补码

1) 原码。按上所述, 正数的符号位用“0”表示, 负数的符号位用“1”表示, 其数值部分随后表示, 称为原码。

仍以上面 N1、N2 为例, 则

$$[N1]_{\text{原}}=0\ 1101001\text{B}$$

$$[N2]_{\text{原}}=1\ 1101001\text{B}$$

原码表示方法简单, 便于与真值进行转换。但在进行减法时, 为了把减法运算转换为加法运算(计算机结构决定了加法运算), 必须引进反码和补码。

2) 反码、补码。在计算机中, 任何有符号数都是以补码形式存储的。对于正数, 其反码、补码与原码相同。例如:

$$N1=+105$$

$$\text{则 } [N1]_{\text{原}}=[N1]_{\text{补}}=[N1]_{\text{反}}=0\ 1101001\text{B}$$

对于负数, 其反码为: 原码的符号位不变, 其数值部分按位取反。例如:

$$N2=-105$$

$$\text{则 } [N2]_{\text{原}}=1\ 1101001\text{B}$$

$$[N2]_{\text{反}}=1\ 0010110\text{B}$$

负数的补码为: 原码的符号位不变, 其数值部分按位取反后再加 1(即负数的反码+1), 称为求补。例如:

$$N2=-105$$

$$\text{则 } [N2]_{\text{补}}=[N2]_{\text{反}}+1$$

$$=1\ 0010110\text{B}+1=1\ 0010111\text{B}$$

如果已知一个负数的补码, 可以对该补码再进行求补码(即一个数的补码的补码), 即得到该数的原码, 即 $[X]_{\text{补}}_{\text{补}}=[X]_{\text{原}}$, 而求出真值。例如:

$$\text{已知: } [N2]_{\text{补}}=1\ 0010111\text{B}$$

$$[N2]_{\text{原}}=11101000\text{B}+1=11101001\text{B}$$

可得真值: $N2=-105$

对采用补码形式表示的数据进行运算时, 可以将减法转换为加法。例如:

设 $X=10$, $Y=20$, 求 $X-Y$ 。

$X-Y$ 可表示为 $X+(-Y)$, 即 $10+(-20)$ 。

$$[X]_{\text{原}}=[X]_{\text{反}}=[X]_{\text{补}}=00001010\text{B}$$

$$[-Y]_{\text{原}}=10010100\text{B}$$

$$[-Y]_{\text{补}}=[-Y]_{\text{反}}+1=11101011\text{B}+1=11101100\text{B}$$

$$\text{则有 } [X+(-Y)]_{\text{补}}=[X]_{\text{补}}+[-Y]_{\text{补}}$$

$$=00001010\text{B}+11101100\text{B} \quad (\text{按二进制相加})$$

$$=11110110\text{B} \quad (\text{和的补码})$$

再对 $[X+(-Y)]_{\text{补}}$ 求补码可得 $[X+(-Y)]_{\text{原}}$, 即:

$$[X+(-Y)]_{\text{原}}=10001001\text{B}+1=10001010\text{B}$$

则 $X-Y$ 的真值为 -10D 。