

高等职业教育技能型紧缺人才培养培训工程系列教材



银领工程

# 单片机实用教程

—— 单片机原理 · 汇编语言 · 接口技术

罗学恒 主编



高等教育出版社

高等职业教育技能型紧缺人才培养培训工程系列教材

# 单片机实用教程

——单片机原理·汇编语言·接口技术

罗学恒 主编

高等教育出版社

## 内容提要

本教材是高等职业教育技能型紧缺人才培养培训工程系列教材之一。

本教材以 MCS-51 系列单片机为主, 注意原理介绍和应用并重, 并且从实用的角度介绍了单片机应用方面的内容。笔者多年从事单片机应用技术教学和科研, 试图向学生较好地解答“什么是单片机? 如何学好单片机? 如何应用单片机?” 这 3 个问题, 使学生学习本教材后, 既能掌握单片机的一般原理, 又能掌握单片机的设计技巧, 还能结合本书中的大量实例掌握单片机应用系统的一般设计方法, 尤其是笔者将自己开发的单片机监控系统分解到每一章的“动手做”, 读者在学完每一章后即动手做, 巩固理论的同时提高动手能力。学完本教材后, 一个成功的产品展现在读者面前, 使学生将来走上工作岗位后能很快进入开发单片机应用系统的角色。

为了便于组织教学, 本教材按循序渐进的顺序编排。本教材共分 12 章, 依次为: 单片机基础知识、单片机硬件结构、指令系统、汇编语言、定时器/计数器、串行接口、中断系统、外部存储器扩展、扩展并行接口、键盘/显示器/语音接口、I<sup>2</sup>C 总线、常用驱动部件接口技术等, 覆盖了开发一个小型单片机控制系统的所有知识。

本书可作为高等职业技术学院、本科院校的二级职业技术学院、高等专科学校、成人高校的计算机和电子类相关专业的教材, 也可供相关专业人员学习参考。

## 图书在版编目(CIP)数据

单片机实用教程——单片机原理·汇编语言·接口  
技术/罗学恒主编. —北京: 高等教育出版社, 2006. 9

ISBN 7-04-020482-7

I. 单... II. 罗... III. 单片微型计算机-教材  
IV. TP368.1

中国版本图书馆 CIP 数据核字(2006)第 100246 号

策划编辑 洪国芬 责任编辑 洪国芬 封面设计 张志奇 责任绘图 朱 静  
版式设计 王 莹 责任校对 王 雨 责任印制 宋克学

出版发行 高等教育出版社  
社 址 北京市西城区德外大街 4 号  
邮政编码 100011  
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司  
印 刷 北京人卫印刷厂

开 本 787×1092 1/16  
印 张 14.75  
字 数 350 000

购书热线 010-58581118  
免费咨询 800-810-0598  
网 址 <http://www.hep.edu.cn>  
<http://www.hep.com.cn>  
网上订购 <http://www.landaco.com>  
<http://www.landaco.com.cn>  
畅想教育 <http://www.widedu.com>

版 次 2006 年 9 月第 1 版  
印 次 2006 年 9 月第 1 次印刷  
定 价 18.80 元

本书如有缺页、倒页、脱页等质量问题, 请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 20482-00

# 出版说明

为了认真贯彻《国务院关于大力推进职业教育改革与发展的决定》，落实《2003—2007年教育振兴行动计划》，缓解国内劳动力市场技能型人才紧缺现状，为我国走新型工业化道路服务，自2001年10月以来，教育部在永州、武汉和无锡连续三次召开全国高等职业教育产学研经验交流会，明确了高等职业教育要“以服务为宗旨，以就业为导向，走产学研结合的发展道路”，同时明确了高等职业教育的主要任务是培养高技能人才。这类人才，既要能动脑，更要能动手，他们既不是白领，也不是蓝领，而是应用型白领，是“银领”。从而为我国高等职业教育的进一步发展指明了方向。

培养目标的变化直接带来了高等职业教育办学宗旨、教学内容与课程体系、教学方法与手段、教学管理等诸多方面的改变。与之相应，也产生了若干值得关注与研究的新课题。对此，我们组织有关高等职业院校进行了多次探讨，并从中遴选出一些较为成熟的成果，组织编写了“银领工程”丛书。本丛书围绕培养符合社会主义市场经济和全面建设小康社会发展要求的“银领”人才的这一宗旨，结合最新的教改成果，反映了最新的职业教育工作思路和发展方向，有益于固化并更好地推广这些经验和成果，很值得广大高等职业院校借鉴。我们的这一想法和做法也得到了教育部领导的肯定，教育部副部长吴启迪专门为首批“银领工程”丛书提笔作序。

我社出版的高等职业教育各专业领域技能型人才培养培训工程系列教材也将陆续纳入“银领工程”丛书系列。

“银领工程”丛书适合于高等职业学校、高等专科学校、成人高校及本科院校举办的二级职业技术学院、继续教育学院和民办高校使用。

高等教育出版社

2006年7月

# 前 言

单片机更确切地应称为微控制器,是20世纪70年代中期发展起来的一种功能强、体积小、可靠性高、面向控制和价格低廉的大规模集成电路器件。单片机的开发应用已成为高科技和工程领域的一项重要内容。在发达国家,单片机技术的开发应用发展很快,并取得了明显的经济效益和社会效益。在我国,近几年单片机的应用研究进展很快,特别是在工业控制、通信、智能化仪器仪表、产品自动化、分布式控制系统中已取得了可喜成果。各高等院校也非常重视这门技术的应用和推广,纷纷开设有关单片机课程。

根据教育部高职高专应用型人才培养目标精神,为满足高职高专单片机实践能力培养的需要,我们编写了本教材,教材涵盖单片机原理、汇编语言、外部存储器、接口技术、驱动等内容。

本教材立足高职高专人才培养目标,遵循主动适应社会发展需要,突出应用性和针对性,着重加强实践能力、应用能力的培养原则,根据高职高专的培养特点,以知识够用、动手能力强为出发点,遵循适应性,突出实用性,强调实践性。考虑高职高专学生的知识层面、学习特点,强化学以致用。

笔者根据多年从事单片机教学的观察发现:学生在学完单片机课程后很茫然,缺乏整体概念,单片机能干什么?怎样完成单片机系统设计?开发单片机系统还应具备哪些知识?本教材正是为解除学生的这些困惑而编写的。

本教材共分12章,第1章讲述单片机的基础知识;第2章讲述单片机的结构;第3章讲述单片机的指令系统;第4章讲述汇编语言;第5章讲述单片机的定时器/计数器及其应用;第6章讲述单片机的串行通信及应用;第7章讲述单片机的中断及其应用;第8章讲述单片机外部存储器扩展;第9章讲述单片机的扩展并行接口;第10章讲述单片机的键盘、显示器和语音接口;第11章讲述I<sup>2</sup>C总线;第12章讲述单片机控制系统的常用驱动部件。

学完本教材后,可使学生基本具备利用单片机开发智能产品的能力。每一章除了备有理论复习的“习题”以外,还配有结合本章知识难点的“动手做”的实验环节,其目的是让学生在掌握理论知识的前提下,强化动手能力。

本教材结合一个实用型单片机最小系统(监控系统)产品,将其分解到每一章的“动手做”环节中,该产品的原理图见附录A,元件清单见附录B。可在学习之前按清单购进元器件,学完一章,在动手做的指导下,将相关器件焊接在印制电路板上完成相关实验(提示:MCS-51单片机在印制电路板上应使用插座,便于经常拔插芯片写程序),一则加深重点难点知识的理解,二则体验实践的乐趣,使学生有一种成就感,提高学习兴趣。等到全书学完,一个单片机最小实用系统也随之做成。学生还可以借助这一平台,展开想象,只需修改程序便可将其改造成不同的智能系统。这样,既解决了教师为设计课犯难的同时,也为将来学生日后的毕业设计提供了更多的课题。

由于第8、第9章不属于最小系统范畴,这两章的“动手做”需另外备用小实验板完成

实验。

各章节中，为便于读者加深理解，加进了许多实用小程序，这些程序是经过多次实验才得到完善的，读者在开发时可作为部分功能块调用。

为方便读者使用，本书配有电子教案。所有书中程序源代码、动手做程序代码、习题答案及电子教案均可登录高等教育出版社高职高专教学资源网 <http://hv.hep.com.cn> 免费下载。

本教材由罗学恒主编，罗懿、龙莎参编，陈晴副教授、杨根兴副教授在百忙中为本教材审稿，在此一并表示感谢！

由于时间仓促、编者水平有限，书中难免还存在疏漏和不足之处，殷切希望广大读者批评指正。

编 者

2006年6月

# 目 录

<b>第 1 章 单片机基础知识</b> .....1	<b>第 3 章 MCS-51 单片机指令系统</b> .....26
1.1 概述.....1	3.1 MCS-51 单片机指令系统简介.....26
1.1.1 什么是单片机.....1	3.1.1 指令格式.....26
1.1.2 单片机的发展.....1	3.1.2 MCS-51 单片机的助记符语言.....27
1.1.3 单片机的特点.....1	3.2 MCS-51 单片机的寻址方式.....27
1.1.4 单片机的应用领域.....1	3.2.1 寻址方式中常用符号注释.....27
1.1.5 单片机的发展趋势.....2	3.2.2 寻址方式.....28
1.1.6 单片机的展望.....2	3.3 数据传送类指令.....32
1.2 数值的转换.....3	3.4 算术运算类指令.....35
1.2.1 数制.....3	3.5 逻辑操作类指令.....39
1.2.2 数制之间的转换.....4	3.6 控制转移类指令.....41
1.3 机器数.....5	3.7 位操作类指令.....45
1.3.1 原码.....5	动手做.....47
1.3.2 反码.....6	习题 3.....48
1.3.3 补码.....6	<b>第 4 章 汇编语言程序设计</b> .....50
1.3.4 无符号数.....7	4.1 汇编语言的格式.....50
1.3.5 BCD 码.....7	4.1.1 伪指令.....50
1.3.6 字符数据编码.....8	4.1.2 汇编语言的格式.....52
1.3.7 内存中的数据.....8	4.2 汇编语言程序设计.....53
习题 1.....9	4.2.1 顺序结构程序.....53
<b>第 2 章 MCS-51 单片机的结构</b> .....10	4.2.2 分支结构程序.....54
2.1 MCS-51 单片机的特点.....10	4.2.3 循环结构程序.....55
2.2 MCS-51 单片机的硬件结构.....10	4.2.4 子程序设计.....58
2.2.1 MCS-51 单片机芯片引脚介绍.....11	4.3 实用程序举例.....59
2.2.2 MCS-51 单片机总体结构.....12	4.3.1 代码转换类程序.....59
2.2.3 MCS-51 单片机存储器.....14	4.3.2 运算类程序.....63
2.2.4 MCS-51 单片机的特殊功能寄存器.....16	动手做.....78
2.2.5 MCS-51 单片机的输入/输出端口.....17	习题 4.....79
2.2.6 MCS-51 单片机复位电路.....19	<b>第 5 章 定时器/计数器</b> .....80
2.2.7 MCS-51 单片机振荡器.....22	5.1 定时器/计数器工作方式及控制.....80
2.2.8 CPU 时序.....22	5.1.1 工作方式寄存器 TMOD.....80
动手做.....24	5.1.2 控制寄存器 TCON.....81
习题 2.....25	5.2 定时器/计数器的工作方式.....82

5.2.1 工作方式 0	82	7.2.1 中断请求源	107
5.2.2 工作方式 1	83	7.2.2 中断控制	108
5.2.3 工作方式 2	83	7.2.3 中断的响应条件及响应过程	109
5.2.4 工作方式 3	83	7.2.4 中断响应时间	110
5.3 定时器/计数器应用举例	84	7.3 中断系统应用举例	110
动手做	85	动手做	113
习题 5	86	习题 7	114
<b>第 6 章 串行接口</b>	<b>87</b>	<b>第 8 章 MCS-51 单片机外部存储器</b>	<b>115</b>
6.1 概述	87	扩展	115
6.1.1 通信方向	87	8.1 存储器扩展概述	115
6.1.2 异步通信和同步通信	87	8.1.1 扩展总线	115
6.1.3 波特率	88	8.1.2 片选信号与地址分配的关系	116
6.1.4 传送编码	88	8.1.3 扩展存储器的步骤	120
6.1.5 信号的调制与调解	89	8.2 扩展程序存储器	121
6.2 串行口工作原理	89	8.2.1 常用程序存储器芯片介绍	121
6.2.1 串行口控制寄存器 SCON	89	8.2.2 EPROM 与单片机的连接	125
6.2.2 电源控制寄存器 PCON	90	8.2.3 扩展 EEPROM 程序存储器	127
6.3 波特率设计	90	8.3 扩展数据存储器	128
6.3.1 方式 0 的波特率	91	8.3.1 常用数据存储器芯片	128
6.3.2 方式 2 的波特率	91	8.3.2 RAM 与单片机的连接	129
6.3.3 方式 1 和方式 3 的波特率	91	8.3.3 扩展数据与程序兼用的存储器	130
6.4 串行通信的 4 种工作方式	92	动手做	131
6.4.1 工作方式 0	92	习题 8	133
6.4.2 工作方式 1	93	<b>第 9 章 MCS-51 单片机扩展并行接口</b>	<b>134</b>
6.4.3 工作方式 2	93	9.1 8255 可编程并行 I/O 接口芯片简介	134
6.4.4 工作方式 3	93	9.1.1 8255 引脚	134
6.5 串行口应用举例	94	9.1.2 8255 工作方式选择	136
6.5.1 利用串行口工作方式 0 扩展 I/O 口	94	9.1.3 8255 应用举例	137
6.5.2 用串行口进行异步单工通信	95	9.2 8155 可编程并行 I/O 接口芯片简介	138
6.5.3 用串行口进行异步双工通信	96	9.2.1 8155 引脚	138
动手做	102	9.2.2 8155 基本功能	139
习题 6	104	9.2.3 8155 应用举例	141
<b>第 7 章 中断系统</b>	<b>105</b>	9.3 简单 I/O 接口芯片简介	143
7.1 中断的概念	105	9.3.1 用锁存器 74LS374 扩展	
7.1.1 为什么要用中断	105	8 位输出口	143
7.1.2 中断源	105	9.3.2 用锁存器 74LS377 扩展	
7.1.3 中断系统的功能	105	8 位输出口	144
7.2 MCS-51 单片机中断系统	106	9.3.3 用锁存器 74LS373 扩展	
		8 位输入口	145



9.3.4 用三态门 74LS244 扩展 8 位输入口 .....	146	11.3.2 读操作模式 .....	188
动手做 .....	147	11.4 单片机与 AT24C×× 接口应用 简单实例 .....	188
习题 9 .....	149	动手做 .....	193
<b>第 10 章 键盘、显示器、语音接口</b> .....	150	习题 11 .....	196
10.1 键盘接口 .....	150	<b>第 12 章 常用驱动部件接口技术</b> .....	197
10.1.1 键盘状态输入与消抖 .....	150	12.1 开关量驱动接口 .....	197
10.1.2 独立式键盘及其与单片机的接口 .....	151	12.1.1 专用集成电路驱动接口电路 .....	197
10.1.3 行列式键盘及其与单片机的接口 .....	153	12.1.2 功率三极管驱动接口电路 .....	198
10.2 显示器接口 .....	160	12.1.3 继电器驱动接口电路 .....	200
10.2.1 LED 显示器的结构与工作原理 .....	161	12.2 光电隔离接口 .....	201
10.2.2 静态显示器接口 .....	163	12.2.1 光电耦合器件 .....	202
10.2.3 动态显示器接口 .....	164	12.2.2 输入通道中的光电隔离 .....	203
10.2.4 LCD 显示器接口技术 .....	167	12.2.3 输出通道中的光电隔离 .....	205
10.3 语音接口 .....	174	12.3 步进电动机的驱动与接口 .....	206
10.3.1 语音芯片 .....	174	12.3.1 步进电动机工作方式 .....	206
10.3.2 语音芯片与单片机的接口 .....	176	12.3.2 步进电动机控制系统和控制方法 .....	207
动手做 .....	178	12.3.3 步进电动机的驱动接口电路 .....	209
习题 10 .....	181	12.4 晶闸管整流器的驱动与接口 .....	211
<b>第 11 章 I<sup>2</sup>C 总线</b> .....	183	12.4.1 单向晶闸管控制电路 .....	211
11.1 I <sup>2</sup> C 器件 AT24C×× 芯片简介 .....	183	12.4.2 双向晶闸管驱动接口 .....	213
11.2 AT24C×× 工作原理 .....	185	动手做 .....	215
11.2.1 开始和结束信号 .....	185	习题 12 .....	216
11.2.2 I <sup>2</sup> C 总线数据 .....	185	<b>附录 A 实用型单片机监控系统</b> 电路连线图 .....	217
11.2.3 I <sup>2</sup> C 数据格式 .....	186	<b>附录 B 实用型单片机监控系统元件表</b> .....	218
11.2.4 I <sup>2</sup> C 总线寻址 .....	186	<b>附录 C MCS-51 系列单片机指令表</b> .....	219
11.3 操作模式 .....	187	<b>参考文献</b> .....	223
11.3.1 写操作模式 .....	187		

---

# 第 1 章 单片机基础知识

---

## 1.1 概述

### 1.1.1 什么是单片机

单片机是单片微型计算机 (Single Chip Microcomputer) 的简称。

单片机是把微型计算机的各个功能部件 (中央处理器 CPU、随机存取存储器 RAM、只读存储器 ROM、I/O 接口、定时器/计数器以及串行通信接口) 集成在一块芯片上, 构成一个完整的计算机, 即单片微型计算机。

由于单片机的结构和功能均是按工业控制要求设计的, 所以, 确切地说, 单片机是单片微型控制器 (Single Chip Microcontroller)。

### 1.1.2 单片机的发展

- ① 1971 年 Intel 公司首次推出 4 位机。
- ② 1976 年 Intel 公司推出 8 位机。
- ③ 1980 年 Intel 公司推出 MCS-51 单片机, 其后 Intel、Philips、SIEMENS、ATMEL 等公司相继推出名目繁多的单片机。
- ④ 1983 年 Intel 公司推出 16 位机。
- ⑤ 20 世纪末 32 位单片机已进入使用阶段。

### 1.1.3 单片机的特点

与 PC 机不同的是单片机的 CPU、RAM、ROM、PIO、SIO、时钟、定时器/计数器等电路集成在一块芯片上, 因此单片机具有集成度高、体积小、功耗低、成本低廉、控制能力强、速度快、抗干扰能力强、易开发等诸多优点。

### 1.1.4 单片机的应用领域

单片机主要用于控制系统, 它的应用领域遍及各行各业, 大到航天飞机, 小至日常生活, 单片机都可以大显其能。单片机在国内的下列五大领域中应用得十分广泛:

第一是家用电器业, 如全自动洗衣机、彩色电视机、电冰箱、VCD、DVD、空调、灯光控制、玩具、智能电子时钟、分时段电度表等都需要单片机来控制。

第二是通信业, 包括电话、IC 卡、手机、中继器、路由器、连接器、终端等都是用单片机作为核心的。

第三个领域是仪器仪表和计算机外设制造,如智能转速计、智能温度计、智能压力计、智能液体流量计、智能频率计、智能位移计以及各种智能传感器和软盘、硬盘、光盘、USB、Hub、打印机、键盘以及票据打印机、计价器、可编程控制器等也都属于这个范畴。

第四是工业控制领域,包括各行各业各种生产设备的自动控制系统。

第五是汽车电子工业,在国外也是单片机应用得十分广泛的一个领域。

除此以外,在管理领域也广泛使用单片机。

### 1.1.5 单片机的发展趋势

#### 1. 低功耗 CMOS 化

MCS-51 系列机功耗为 630mW,现行单片机普遍采用 CMOS(互补金属氧化物半导体工艺)和 HMOS(高密度互补金属氧化物半导体工艺)以及 CHMOS(互补高密度金属氧化物半导体工艺),其功耗均约为 100mW,时钟频率为 16 MHz。由于功耗低,非常适合由电池供电。

#### 2. 多功能微型化

常规的单片机通常都是将中央处理器(CPU)、RAM、ROM、PIO、SIO、中断系统、定时电路、时钟电路集成在一块芯片上的。而现行单片机在原来基础上还将 A/D、D/A、PMW(脉冲调制电路)、WTD(看门狗)、红外接口、LED、LCD 驱动电路集成在一块芯片上,且体积比常规单片机还要小。有些厂商还可以根据用户要求量身定做。

#### 3. 主流和多品种共存

① 主流机仍是以 80C51 为核心的各种单片机。与其结构和指令系统兼容的有 Philips、ATMEL、Winbond 等公司的产品。

② 与 80C51 不兼容的有 MicoCHIP、Motorola、HOLTEK 等公司的产品。

③ 还有像日本的三菱、东芝、松下、富士通等公司的专用单片机。

### 1.1.6 单片机的展望

在 20 世纪 50、60 年代,最具代表性的先进的电子技术就是无线电技术,包括无线电广播、收音机、无线通信(电报)、业余无线电台、无线电定位、导航等遥测、遥控、遥信技术。早期就是这些电子技术带领着许多青少年步入了奇妙的电子世界,无线电技术展示了当时科技生活美妙的前景。电子科学逐渐形成了一门新型学科。无线电电子学、无线通信开始了电子世界的历程。

单片机是器件级计算机系统,它可以嵌入到任何对象体系中去,实现智能化控制。它将以嵌入式系统作为主干形成最富活力的新型学科。

正如绿色食品深受人们青睐一样,任何引入单片机的智能产品将备受人们关注。

如果说 20 世纪 50 年代起,无线电世界造就了几代精英,那么当今的单片机世界将会造就出新一代电子精英。

## 1.2 数值的转换

计算机作为信息处理的工具，数据表示的问题是计算机工作的出发点。计算机内部基于硬件处理简单的原因，采用二进制（Binary）作为数据表示的基础，日常生活中习惯使用十进制（Decimal）数据，汇编语言程序中喜欢使用十六进制（Hexadecimal）数据。

### 1.2.1 数制

#### 1. 二进制数

二进制数由 0、1 两个数码构成，基数为 2，第  $i$  位的权为  $2^i$ ，运算时遵循“逢 2 进 1”的原则。书写时在尾部加注字母 B 或下标 2 描述，如 1101B 或  $(1101)_2$ 。其他进制数的书写加注相应的字母或下标描述，缺省一般指十进制数。

二进制数  $a_n a_{n-1} \cdots a_0 . b_{-1} b_{-2} \cdots b_{-m}$  的值是：

$$a_n \times 2^n + a_{n-1} \times 2^{n-1} + \cdots + a_0 + b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + \cdots + b_{-m} \times 2^{-m}$$

其中  $a_i$  和  $b_i$  为 0、1 两个数码中的一个。

**【例 1-1】**  $101011\text{B} = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 43\text{D}$

$$100000000\text{B} = 1 \times 2^8 = 256\text{D}$$

$n$  位二进制数可以表示  $2^n$  种组合。4 位二进制数能表示 16 种组合（0~15 的整数），8 位二进制数能表示 256 种组合（0~255 的整数），16 位二进制数能表示  $64 \times 1024$  种组合（0~65 535 的整数）。

#### 2. 十六进制数

使用二进制数在计算机中容易实现，便于存储，抗干扰性强。但位数较多时，阅读、书写、记忆、输入等不太方便。

若干年前，程序员发现操作时一般是对位的“组”，而不是个别的位的操作。最早的微处理器 4 位，自然想到用 4 位一组缩写二进制数，即十六进制数。

十六进制的数由 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 十六个数码构成，其中数码 A、B、C、D、E、F 对应于十进制数 10、11、12、13、14、15，基数为 16，第  $i$  位的权为  $16^i$ ，运算时遵循“逢 16 进 1”的原则。

书写时在尾部加注字母 H 或下标 16 描述，如 1FH 或  $(1\text{F})_{16}$ 。

描述以字母开头的十六进制数前面应加一数码 0，以与标识符（汇编语言中详细介绍）相区别（如 0FAH）。

十六进制数  $a_n a_{n-1} \cdots a_0 . b_{-1} b_{-2} \cdots b_{-m}$  的值是：

$$a_n \times 16^n + a_{n-1} \times 16^{n-1} + \cdots + a_0 \times 16^0 + b_{-1} \times 16^{-1} + b_{-2} \times 16^{-2} + \cdots + b_{-m} \times 16^{-m}$$

其中  $a_i$  和  $b_i$  为 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 十六个数码中的一个。

**【例 1-2】**  $1011\text{H} = 1 \times 16^3 + 0 \times 16^2 + 1 \times 16^1 + 1 \times 16^0 = 4113\text{D}$

$$14\text{AFH} = 1 \times 16^3 + 4 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 = 5295\text{D}$$

汇编语言调试系统中的数全部使用十六进制数表示，汇编列表文件中代码指令及内存地址也使用十六进制数表示。可以说十六进制数是汇编语言的书写工具，应尽快习惯使用。

### 3. 八进制数

计算机中一个字节是 8 位，也常用八进制数来缩写二进制数。

八进制数由 0、1、2、3、4、5、6、7 八个数码构成，基数为 8，第  $i$  位数的权为  $8^i$ ，运算时遵循“逢 8 进 1”的原则。

书写时在尾部加注字母 O 或下标 8 描述，如 76O 或  $(76)_8$ 。

由于字母 O 与数字 0 容易混淆，八进制数亦常用尾标 Q 标识，如 76Q。

八进制数  $a_n a_{n-1} \cdots a_0 . b_{-1} b_{-2} \cdots b_{-m}$  的值是：

$$a_n \times 8^n + a_{n-1} \times 8^{n-1} + \cdots + a_0 \times 8^0 + b_{-1} \times 8^{-1} + b_{-2} \times 8^{-2} + \cdots + b_{-m} \times 8^{-m}$$

其中  $a_i$  和  $b_i$  为 0、1、2、3、4、5、6、7 八个数码中的一个。

【例 1-3】  $4056Q = 4 \times 8^3 + 0 \times 8^2 + 5 \times 8^1 + 6 \times 8^0 = 2094D$

$1001Q = 1 \times 8^3 + 0 \times 8^2 + 0 \times 8^1 + 1 \times 8^0 = 513D$

## 1.2.2 数制之间的转换

### 1. 非十进制数与十进制数之间的转换

十进制数转换成非十进制数，转换时将整数部分与小数点部分分别转换。整数部分采用除基数取余法，直至商为 0，先得到的余数为低位，后得到的余数为高位。小数部分采用乘基数取整法，直至乘积为整数或达到控制精度。

【例 1-4】  $17D = 10001B$  ;  $17/2 = 8 \cdots 1$ ,  $8/2 = 4 \cdots 0$ ,  $4/2 = 2 \cdots 0$ ,  $2/2 = 1 \cdots 0$ ,  $1/2 = 0 \cdots 1$

$57D = 71Q$  ;  $57/8 = 7 \cdots 1$ ,  $7/8 = 0 \cdots 7$

$57D = 39H$  ;  $57/16 = 3 \cdots 9$ ,  $3/16 = 0 \cdots 3$

【例 1-5】  $0.625D = 0.101B$  ;  $0.625 \times 2 = 1.25 \cdots 1$ ,  $0.25 \times 2 = 0.50 \cdots 0$ ,  $0.5 \times 2 = 1 \cdots 1$

$0.625D = 0.5Q$  ;  $0.625 \times 8 = 5 \cdots 5$

$0.625D = 0.AH$  ;  $0.625 \times 16 = 10 \cdots A$

非十进制数转换成十进制数的例子参见例 1-1 至例 1-3。

### 2. 二、八、十六进制数之间的转换

二、八、十六进制数之间的转换可以先将其转换成十进制数，再将对应的十进制数转换成需要转换的进制数。

由于二、八、十六进制数之间的特殊关系，还可以采用分组法快速转换。将二进制数转换成八进制数可按 3 位一组进行分组，转换成十六进制数可按 4 位一组进行分组，每一组对应八进制或十六进制的相应数码，参见表 1-1，组合即得转换结果。分组时如果位数不够，整数部分在最左边补 0，小数部分在最右边补 0。

将八进制数转换成二进制数，只需将八进制数的每一位对应转换成 3 位二进制数即可。同样将十六进制数转换成二进制数，只需将其每一位对应转换成 4 位二进制数即可。

表 1-1 二、八、十六、十进制数对应关系表

十进制数	二、八进制数对应关系		二、十六进制数对应关系	
	二进制数	八进制数	二进制数	十六进制数
0	000	0	0000	0
1	001	1	0001	1
2	010	2	0010	2
3	011	3	0011	3
4	100	4	0100	4
5	101	5	0101	5
6	110	6	0110	6
7	111	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

【例 1-6】  $1100100.11010B=001\ 100\ 100.110\ 100=144.64Q$

$1100100.11010B=0110\ 0100.1101\ 0000=64.D0H$

反过来，八进制数 144.64Q、十六进制数 64.D0H 转换成二进制数即为 1100100.1101B。

## 1.3 机器数

数的原值称为真值，是计算机中表示数的实际数值。

数在计算机中的二进制表示形式称为机器数。机器数可以用不同的码制表示，常用的有原码、反码和补码三种，广泛使用的是原码和补码两种。

为了便于理解，常将数用中括号括起来，在尾部加注下标原、反、补来明确码制。

机器数将数的符号也数字化，一般用最高位表示数的符号，“0”代表正数，“1”代表负数。

机器数的表示形式还与存储位数有关，也就是说，数据在计算机中的存储方式是以一个字（8 位二进制数）作为一个基本存储单元的，超出 8 位的二进制数则需要 2 个字节、3 个字节、4 个字节甚至更多。最常用的是单字节、2 字节和 4 字节。

【例 1-7】 数+1010B，相应机器数为 00001010B（以单字节表示）。

数-1010B，相应机器数为 10001010B（以单字节表示）。

数+100000010B，相应机器数为 0000000100000010B（以 2 字节表示）。

数-100000010B，相应机器数为 1000000100000010B（以 2 字节表示）。

### 1.3.1 原码

原码表示数是一种比较直观的机器数的表示方法，原码与真值的区别仅仅是数的符号数字化，符号数字化的二进制数称为原码。

**【例 1-8】**  $X=+1011\text{B}$ ,  $[X]_{\text{原}}=00001011$

$Y=-1011\text{B}$ ,  $[Y]_{\text{原}}=10001011$

原码表示中, 最高位为符号位, 其他位表示数的绝对值。

$n$  位原码表示的数的范围为  $[-(2^{n-1}-1), +2^{n-1}-1]$ 。8 位原码表示的数的范围是  $[-127, 127]$ , 16 位原码表示的数的范围是  $[-32\ 767, 32\ 767]$ 。

原码表示数简单直观, 做乘除运算比较方便, 可取绝对值直接运算, 而符号单独处理。但对于应用得最多的加减运算, 原码表示不太方便, 像  $(-2)+3$  表面上进行加法操作, 而实际上是进行  $(3-2)$  减法操作。

### 1.3.2 反码

正数的反码与原码相同, 负数的反码等于其原码 (除符号位以外) 按位取反, 也等于其对应正数的原码 (含符号位) 按位取反。

**【例 1-9】**  $X=+1011\text{B}$ ,  $[X]_{\text{反}}=00001011$

$Y=-1011\text{B}$ ,  $[Y]_{\text{反}}=11110100$

反码表示数的范围与原码相同。

### 1.3.3 补码

补码表示能让符号位一同参与数的运算, 能将减法转化为加法进行运算。

计算机的补码来源于数学中的“模”运算, 是以  $2^n$  为模, 基于 2 的补码。

正数的补码等于其本身, 与其原码相同。负数的补码等于其原码 (除符号位以外) 按位取反, 末位加 1。

**【例 1-10】**  $X=+1011\text{B}$ ,  $[X]_{\text{补}}=00001011$

$Y=-1011\text{B}$ ,  $[Y]_{\text{补}}=11110101$

负数的补码也等于其对应正数的补码 (含符号位) 按位取反, 末位加 1, 也就是反码末位加 1。

$[X]_{\text{补}}+[Y]_{\text{补}}=[X+Y]_{\text{补}}$  符号位同时参加运算, 只要不溢出, 补码的和等于和的补码。

$[X-Y]_{\text{补}}=[X]_{\text{补}}+[-Y]_{\text{补}}$  减法运算转化成加法运算, 符号位同时参加运算。

如何从补码求原码或真值, 可对补码再一次求补码得到, 即  $[[X]_{\text{补}}]_{\text{补}}=[X]_{\text{原}}$ 。其实对于正数, 补码的真值就等于其自身。

$n$  位补码表示的数的范围为  $[-2^{n-1}, 2^{n-1}-1]$ 。为了扩大数的表示范围, 可用多个字节来表示一个数。8 位补码 (1 个字节) 表示的数的范围为  $[-128, 127]$ , 16 位补码 (2 个字节) 表示的数的范围为  $[-32\ 768, 32\ 767]$ 。

在计算机中, 负数用补码表示, 符号数用补码表示。

表 1-2 给出了 8 位二进制数的原码、补码及反码表示。

表 1-2 8 位二进制数的编码表示

二进制数	原 码	补 码	反 码
00000000	+0	+0	+0
00000001	+1	+1	+1

续表

二进制数	原 码	补 码	反 码
00000010	+2	+2	+2
...	...	...	...
01111110	+126	+126	+126
01111111	+127	+127	+127
10000000	-0	-128	-127
10000001	-1	-127	-126
...	...	...	...
11111101	-125	-3	-2
11111110	-126	-2	-1
11111111	-127	-1	-0

综上所述,不难看出:机器数的形式值不一定等于真值,但与真值肯定存在某种对应关系。

也就是说,无符号数(可理解为单纯的正数)的机器数等于真值,而有符号数的机器数是真值的绝对值的补码。

对于正数来说:机器数  $X=[X]_{\text{原}}=[X]_{\text{补}}$ ;

对于负数来说:机器数  $X \neq [X]_{\text{补}} \neq [X]_{\text{原}}$ 。

### 1.3.4 无符号数

在某些情况下,要处理的数据全是正数,此时保留符号位毫无意义。如将符号位也作为数据位处理,可形成无符号数。

【例 1-11】10011001B,表示无符号整数是  $1 \times 2^7 + 1 \times 2^4 + 1 \times 2^3 + 1$ ,即 153

表示有符号整数是  $-(1 \times 2^4 + 1 \times 2^3 + 1)$ ,即 -25

$n$  位无符号整数的范围为  $0 \sim 2^n - 1$ ,8 位(一个字节)无符号整数范围为  $0 \sim 255$ ,16 位(2 个字节)无符号整数范围为  $0 \sim 65535$ 。

在计算机中,用无符号数表示存储空间的地址。

### 1.3.5 BCD 码

BCD 码(Binary Coded Decimal)使用 4 位二进制数来表示 1 位十进制数,常称为二进制编码的十进制数。

4 位二进制数能表示 16 种状态,可用其中任意 10 种状态表示十进制数字  $0 \sim 9$ ,由此形成 8421 码、2421 码、余 3 码等多种 BCD 码,最常用的是 8421 码。

8421 码的编码方法见表 1-3,8421 是指用于编码的 4 位二进制各位的权,由此也不难推出 2421 码的编码方法。

表 1-3 8421 码编码表

十进制	0	1	2	3	4	5	6	7	8	9
二进制	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

【例 1-12】1329D 用 BCD 码表示为:0001 0011 0010 1001



67536D 用 BCD 码表示为：0110 0111 0101 0011 0110

BCD 码的使用为十进制在计算机内的表示提供了一种简单的手段，计算机提供了直接处理 BCD 码的一组指令，但 BCD 码参与运算较复杂，这些指令在实际编程时很少使用。

十进制数的 BCD 码表示并不是十进制数通过数制转换形成的二进制数。

BCD 码有压缩 BCD 码和非压缩 BCD 码两种不同的存储方式。压缩 BCD 码用 1 个字节存放 2 位十进制数字，每位十进制数字占 4 位，一位十进制数字占低 4 位，另一位十进制数字占高 4 位；而非压缩 BCD 码用 1 个字节存放 1 位十进制数字，每位十进制数字占每个字节的低 4 位，高 4 位一般为 0。

【例 1-13】 9329D 用 BCD 码表示为：1001 0011 0010 1001

非压缩 BCD 码存放需 4 个字节，压缩 BCD 码存放只需 2 个字节。

### 1.3.6 字符数据编码

字符数据包括字母、数字、专用字符及一些控制字符。字符数据的使用为人和计算机之间交换信息提供了方便，美国信息交换标准代码 ASCII 码（American Standard Code Information Interchange）是最主要的字符编码方式。

ASCII 码采用 1 个字节的低 7 位进行编码，能表示 128 个字符。最高位用做奇偶校验位。表 1-4 用十六进制形式给出了主要字符的 ASCII 码。数字 0~9 的 ASCII 码为 30H~39H，大写字母 A~Z 的 ASCII 码为 41H~5AH，小写字母 a~z 的 ASCII 码为 61H~7AH，控制字符的 ASCII 码为 00H~1FH 及 7FH，专用字符的 ASCII 码散列其中。

表 1-4 主要字符的十六进制 ASCII 码表

字符	ASCII	字符	ASCII	字符	ASCII	字符	ASCII	字符	ASCII	字符	ASCII
NUL	00	+	2B	;	3B	K	4B	[	5B	k	6B
BEL	07	,	2C	<	3C	L	4C	\	5C	l	6C
LF	0A	-	2D	=	3D	M	4D	]	5D	m	6D
FF	0C	/	2E	>	3E	N	4E	↑	5E	n	6E
CR	0D	。	2F	?	3F	O	4F	←	5F	o	6F
SP	20	0	30	@	40	P	50	,	60	p	70
!	21	1	31	A	41	Q	51	a	61	q	71
"	22	2	32	B	42	R	52	b	62	r	72
#	23	3	33	C	43	S	53	c	63	s	73
\$	24	4	34	D	44	T	54	d	64	t	74
%	25	5	35	E	45	U	55	e	65	u	75
&	26	6	36	F	46	V	56	f	66	v	76
'	27	7	37	G	47	W	57	g	67	w	77
(	28	8	38	H	48	X	58	h	68	x	78
)	29	9	39	I	49	Y	59	i	69	y	79
*	2A	:	3A	J	4A	Z	5A	j	6A	z	7A

### 1.3.7 内存中的数据

数据在计算机内部采用何种方式，依赖于程序执行的情况，可用二进制、BCD 码与字符方