

赠送
电子课件

单片机原理与应用 及C51程序设计 (第3版)

谢维成 杨加国 主编
董秀成 主审



清华大学出版社

单片机原理与应用及 C51 程序设计 (第 3 版)

谢维成 杨加国 主 编



清华大学出版社
北京

内 容 简 介

MCS-51 系列单片机应用广泛,是学习单片机技术较好的系统平台,同时也是单片微型计算机系统开发的一个重要系列。本书以实用为宗旨,通过丰富的实例讲解 MCS-51 单片机原理和软件开发技术,并采用对比方法,同一功能分别以单片机汇编语言程序和单片机 C 语言程序实现,且附有 Proteus 仿真,免费提供所有源代码和电路图的资源下载。

本书为原书第 3 版,共 11 章。

本书适合各类大专院校及培训机构作为“单片机原理与应用”或“单片机应用系统开发”类课程的教材,特别适合打算学习单片机应用系统开发的读者,也可供各类电子工程、自动化技术人员和计算机爱好者参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

单片机原理与应用及 C51 程序设计/谢维成,杨加国主编. —3 版. —北京:清华大学出版社,2014
(2014.7 重印)

ISBN 978-7-302-36715-4

I. ①单… II. ①谢… ②杨… III. ①单片微型计算机—C 语言—程序设计 IV. ①TP368.1 ②TP312

中国版本图书馆 CIP 数据核字(2014)第 115868 号

责任编辑:梁媛媛

封面设计:杨玉兰

责任校对:宋延清

责任印制:刘海龙

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62791865

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:24.5 字 数:596 千字

版 次:2006 年 7 月第 1 版 2014 年 7 月第 3 版 印 次:2014 年 7 月第 2 次印刷

印 数:3001~6000

定 价:39.80 元

前 言

《单片机原理与应用及 C51 程序设计》一书自 2006 年出版后，以其全新的编写思路、鲜明的应用性特色，受到了广大教师及学生的欢迎，重印了 9 次。2009 年修订出版了《单片机原理与应用及 C51 程序设计(第 2 版)》，也重印了 9 次。为了更加适合各院校使用，编者认真听取了广大师生的意见，在《单片机原理与应用及 C51 程序设计(第 3 版)》中，我们仍然保留了第 1、2 版的主体框架和特色，但更加突出应用性，对实例增加了 Proteus 仿真，对相应的内容做了一些修订和增补调整。

MCS-51 系列单片机应用广泛，是学习单片机技术较好的系统平台，同时也是单片微型计算机应用系统开发的一个重要系列。我们承担了四川省高等教育教学改革工程人才培养质量和教学改革项目“微机、单片机、接口技术系列实验及实践教学改革”，提出的实验及实践教学改革的目的是培养应用型人才。根据理论教学和实践教学的经验，发现学生要想熟练地掌握 MCS-51 单片机应用系统软件设计，就必须完全理解单片机汇编语言，只有这样，才能理解并掌握 MCS-51 程序设计。若在用汇编语言讲授单片机原理后另外单独开设一门“MCS-51 程序设计”课程，由于时间间隔的原因，学生往往不能与原理很好地联系起来进行对比学习。因此我们尝试在课堂上讲解单片机原理的同时介绍单片机 C 语言程序设计，避免直到进入实验室或开发实践阶段时才讲授单片机 C 语言程序设计以及开发环境，这样可以为开设综合实验和创新性实验奠定一定的基础。

因此本书的目的是想在讲解单片机基本原理的同时，能兼顾汇编语言和 C 语言两个方面。单片机原理与应用教材大都采用汇编语言讲解和设计程序实例，但汇编语言学习困难。在实际应用系统开发调试中，特别是开发比较复杂的应用系统时，为了提高开发效率和使程序便于移植，现在多用 C 语言。C 语言不仅学习起来容易，而且也同汇编语言一样，能够对单片机资源进行访问，因而目前大多数院校在开设单片机课程时都引入了 C 语言。但引入 C 语言后，在选教材时就发现存在两个方面的问题：第一，单片机原理与应用(含单片机 C 语言程序设计)的教材不多，而兼顾汇编语言和 C 语言的教材更少，所以可选择的余地较小；第二，单片机 C 语言方面的教材一般面向开发，不讲原理，属于高级教程，不适合初学者。而我们需要一本在讲单片机基本原理的同时能兼顾汇编语言和 C 语言两个方面的教材，以避免学生在学习“单片机原理与应用”课程时还要另外参考一本单片机 C 语言方面的教材。

基于前边所讲的原因，我们一直按照“同一实例，汇编语言和 C 语言对比学习”的思路教学并编写教材及改版。在本书的实例中，相同的功能用汇编语言和 C 语言分别实现，通过用汇编语言和 C 语言两个方面的编程对比，使学生能够有选择地掌握一种并认识另一种。对于把“单片机原理与应用”及“MCS-51 程序设计”作为两门课程分别开设的学校，也可以使用同一本教材，对学习者的复习单片机原理及汇编语言知识有很大的帮助。同时，为了提高学生应用设计的能力，还介绍了目前单片机接口常用的接口芯片，列举了几个简单的单片机应用系统开发实例。

1. 第 3 版内容的调整

第 1 章“计算机基础知识”，增加了 1.2 节“微型计算机的基本结构和工作原理”，为没有学习微机原理课程的读者补充相关的基础知识，学了微机原理课程的读者可以不学，另外，1.3 节“单片微型计算机”由以前的 1.2、1.3、1.4 节合并而成。

第 2 章“单片机基本原理”，做了修订。

第 3 章“单片机汇编程序设计”，做了修订；3.1 及 3.5 做了相应的修改；把第 2 版的第 4、5 章合并到第 4 章“单片机 C 语言程序设计”，只保留与标准 C 有区别的地方；根据实际教学流程及实验安排，把第 2 版的第 12 章提前到第 5 章“Keil C51 集成环境的使用”，用汇编程序介绍。

新增了第 6 章“Proteus 软件的使用”，其中 6.2 节“Proteus 的基本操作”以并口应用做介绍。

第 7 章到第 10 章都增加了 Proteus 仿真，根据仿真对内容做了相应的调整；把第 2 版的第 10、11 章合并到第 11 章。

第 11 章“单片机应用系统设计及举例”，也增加了 Proteus 仿真，根据仿真做相应的调整，并增加了 11.4 节“多点温度测量系统的设计”。

2. 本书特点

本书以实用为宗旨，用众多的实例讲解 MCS-51 单片机原理和硬、软件开发技术，针对同一功能，同时提供单片机汇编语言源程序和单片机 C 语言源程序，并实现 Proteus 仿真，免费提供所有源代码和电路图的资源下载。从实用的角度出发，书中配备了大量的实例，详细描述了实例的具体设计步骤，并提供单片机汇编源程序和单片机 C 语言源程序的详细代码，并且完整地阐述了单片机应用系统分析和开发的全过程，读者可以此作为进入单片机应用系统开发领域的首次尝试。

本书与传统的单片机基本原理书籍相比较，更面向实际开发，与单片机 C 语言程序设计书籍相比，兼顾了单片机原理和汇编语言的讲解，同时提供了 Proteus 仿真，有利于初学者迅速掌握单片机技术。

本书图文并茂，实用性强，为便于读者练习和自学，各章均配有少量习题。本书可作为大专院校单片机原理与应用类课程的教材，也可作为单片机原理与应用技术培训班的教材，特别适合打算学习单片机应用系统开发技术的读者，同时可供各类电子工程、自动化技术人员和计算机爱好者参考。

3. 本书内容

本书共分 11 章，具体内容如下。

第 1 章：计算机基础知识。

第 2 章：单片机的基本原理。

第 3 章：单片机汇编程序设计。

第 4 章：单片机 C 语言程序设计。

第 5 章：Keil C51 集成环境的使用。

第 6 章：Proteus 软件的使用。

第 7 章：MCS-51 单片机的内部资源及编程。

第 8 章：MCS-51 单片机常用接口。

第 9 章：D/A、A/D 转换器与 51 单片机的接口。

第 10 章：MCS-51 单片机的其他接口。

第 11 章：单片机应用系统设计及举例。

附录中提供了 MCS-51 系列单片机指令表、C51 运算符优先级和结合规则及 C51 库函数表，以及单片机相关的网络资源列表，以使读者找到更广阔的学习园地。

4. 如何使用本书

对于 MCS-51 单片机的初学者来说，应该从本书的第 1 章开始进行学习，以了解 MCS-51 单片机技术的基本知识和 MCS-51 单片机的使用方法，掌握 MCS-51 单片机结构和相应接口芯片的具体使用方法，以及与 MCS-51 单片机汇编语言编程和单片机 C 语言编程相关的具体技术，学完 1~11 章，即可达到从事单片机应用系统开发的基本要求。

对于已经具有一定 MCS-51 单片机技术基础，比较了解 MCS-51 单片机的读者来说，可以直接从第 4 章开始学习，重点理解和掌握使用 MCS-51 单片机开发应用系统的相关技术，通过对比来掌握单片机汇编语言编程和单片机 C 语言编程的方法，着重掌握单片机应用系统的开发过程。

建议本书的理论课安排在 60 学时左右，实验 16 学时，如果只学习汇编程序设计或 C 语言程序设计，理论学习课时可适当地减少。课程学习后，可安排相应的课程设计，以便对学习内容进行巩固和加深理解。

另外，本书在描述中把 MCS-51 单片机常简称为“51 单片机”，书中采用了 Keil μ Vision 4 软件界面，读者在学习过程中也可以采用 Keil C51 的最新版本，或者可以从本书提供的资源网站中搜索下载其对应的软件包，以供学习和使用。

5. 我们的经验

根据我们的教学和开发经验，学习单片机技术，特别是学习单片机应用系统开发技术时，关键是让学习者自己迅速地找到适合自己的学习方法，必须在第一时间使学习者看到自己的学习成绩，排除“对硬件设计没有信心，畏惧编程”的心理因素。因此有必要走“依葫芦画瓢”的道路，在实验中模拟开发出简单的应用系统，然后逐渐地由浅入深，逐步进入单片机应用系统开发领域。

为此目的，本书给出了大量实例，包括硬件电路设计和应用系统开发，我们希望读者通过大量的实例来加深对相关内容的认识和理解，尽快地把理论知识转换为解决实际问题的能力。另一方面，为方便读者快速阅读本书，书中各实例中的所有源代码和电路图均提供下载，读者可以根据自己的实际情况进行选择和使用，建议读者详细阅读第 5~11 章，并分析电路和程序源代码，最好能够自己在实验室模拟一个单片机应用系统实验项目进行开发练习，以此作为真正的单片机应用系统开发的起步。

6. 致谢

本书由西华大学的谢维成和成都大学的杨加国、陈斌、杨显富共同编写，谢维成和杨加国担任主编。

本书第 1、2、3、9、11 章由杨加国编写，第 4 章和第 7 章由陈斌编写，第 5、6 章由杨显富编写，第 8、10 章和附录由谢维成编写，最后由谢维成和杨加国统稿完成。另外王胜、郑海春、王孝平、杨帆、龙驹、陈永强、李茜、宋玉忠参与了本书部分图形的绘制及仿真调试工作，在此一并表示感谢。同时感谢参考文献的作者们，本书借鉴了他们的部分成果，他们的工作给了我们很大的帮助和启发。

尽管《单片机原理与应用及 C51 程序设计(第 3 版)》是改版，我们全体参编人员已尽心尽力，但限于自身水平，书中难免出现遗漏和错误之处，恳请广大读者不吝指正。

编 者

目 录

第 1 章 基础知识1	
1.1 信息在计算机中的表示.....1	
1.1.1 数在计算机内的表示.....1	
1.1.2 字符在计算机内的表示.....5	
1.2 微型计算机的基本结构和工作原理.....6	
1.2.1 微型计算机的发展6	
1.2.2 微型计算机的基本结构.....7	
1.2.3 微处理器8	
1.2.4 存储器11	
1.2.5 输入/输出设备及 I/O 接口 电路14	
1.2.6 总线16	
1.2.7 微型计算机的工作过程.....17	
1.3 单片微型计算机21	
1.3.1 单片机的基本概念21	
1.3.2 单片机的主要特点22	
1.3.3 单片机的发展及其主要品种.....22	
1.3.4 单片机的应用26	
习题.....27	
第 2 章 单片机的基本原理29	
2.1 51 系列单片机简介29	
2.2 51 系列单片机的结构原理.....29	
2.2.1 51 系列单片机的基本组成.....29	
2.2.2 51 系列单片机的内部结构.....30	
2.2.3 51 系列单片机的中央 处理器(CPU).....31	
2.2.4 51 系列单片机的存储器 结构32	
2.2.5 51 系列单片机的输入/输出 接口39	
2.3 51 系列单片机的外部引脚及 片外总线42	
2.3.1 外部引脚43	
2.3.2 片外总线结构45	
2.4 51 系列单片机的工作方式.....45	
2.4.1 复位方式 45	
2.4.2 程序执行方式..... 46	
2.4.3 单步执行方式..... 46	
2.4.4 掉电和节电方式..... 47	
2.4.5 编程和校验方式..... 48	
2.5 51 系列单片机的时序..... 49	
2.5.1 时钟周期、机器周期和 指令周期 49	
2.5.2 访问外部 ROM 的时序..... 51	
2.5.3 访问外部 RAM 的时序..... 51	
习题 52	
第 3 章 单片机汇编程序设计 53	
3.1 51 系列单片机的汇编指令格式及 标识 53	
3.1.1 指令格式 53	
3.1.2 指令中用到的标识符..... 54	
3.2 51 单片机指令的寻址方式 54	
3.2.1 常数寻址(立即寻址)..... 55	
3.2.2 寄存器数寻址(寄存器寻址).... 55	
3.2.3 存储器数寻址..... 55	
3.2.4 位数据寻址(位寻址)..... 57	
3.2.5 指令寻址 57	
3.3 51 单片机的指令系统..... 58	
3.3.1 数据传送指令..... 58	
3.3.2 算术运算指令..... 61	
3.3.3 逻辑操作指令..... 63	
3.3.4 控制转移指令..... 65	
3.3.5 位操作指令..... 70	
3.4 51 单片机汇编程序常用伪指令 72	
3.5 51 单片机汇编程序设计 76	
3.5.1 数据传送程序..... 76	
3.5.2 运算程序 77	
3.5.3 代码转换程序..... 81	
3.5.4 多分支转移(散转)程序 82	

3.5.5 延时程序	84	5.3.2 如何查看和修改寄存器的 内容	119
习题	84	5.3.3 如何观察和修改变量	119
第 4 章 单片机 C 语言程序设计	87	5.3.4 如何观察存储器区域	119
4.1 C51 基本知识	87	习题	120
4.2 C51 的数据类型	88	第 6 章 Proteus 软件的使用	121
4.2.1 C51 的基本数据类型	89	6.1 Proteus 概述	121
4.2.2 C51 特有的数据类型	90	6.1.1 Proteus 的进入	121
4.3 C51 的变量与存储类型	91	6.1.2 Proteus 的界面	122
4.3.1 C51 的普通变量及定义	91	6.2 Proteus 的基本操作	124
4.3.2 C51 的特殊功能寄存器变量	94	6.2.1 新建电路, 选择元件	124
4.3.3 C51 的位变量	94	6.2.2 放置和调整元件	126
4.3.4 C51 的指针变量	95	6.2.3 连接导线	127
4.4 绝对地址的访问	97	6.2.4 给单片机加载程序	130
4.4.1 使用 C51 运行库中的 预定义宏	97	6.2.5 运行仿真查看结果	131
4.4.2 通过指针访问	98	习题	131
4.4.3 使用 C51 扩展关键字 <code>_at_</code>	98	第 7 章 51 单片机的内部资源及编程	133
4.5 C51 中的函数	99	7.1 并行输入/输出接口	133
4.5.1 C51 函数的参数传递	99	7.2 定时/计数器接口	134
4.5.2 C51 函数的返回值	100	7.2.1 定时/计数器的主要特性	134
4.5.3 C51 函数的存储模式	100	7.2.2 定时/计数器 T0、T1 的 结构及工作原理	134
4.5.4 C51 的中断函数	101	7.2.3 定时/计数器的方式和 控制寄存器	135
4.5.5 C51 函数的寄存器组选择	102	7.2.4 定时/计数器的工作方式	136
4.5.6 C51 的重入函数	102	7.2.5 定时/计数器的初始化 编程及应用	139
习题	103	7.3 串行接口	144
第 5 章 Keil C51 集成环境的使用	105	7.3.1 通信的基本概念	144
5.1 Keil C51 简介	105	7.3.2 MCS-51 单片机串行口的 功能与结构	146
5.1.1 Keil μ Vision 4 IDE 的安装	105	7.3.3 串行口的工作方式	149
5.1.2 Keil μ Vision 4 IDE 界面	105	7.3.4 串行口的编程及应用	151
5.2 Keil μ Vision 4 IDE 的使用方法	111	7.4 中断系统	161
5.2.1 建立项目文件	111	7.4.1 中断的基本概念	161
5.2.2 给项目添加程序文件	113	7.4.2 MCS-51 单片机的 中断系统	162
5.2.3 编译、连接成目标文件	114		
5.2.4 运行调试、观察结果	115		
5.2.5 仿真环境的设置	116		
5.3 Keil C51 的调试技巧	119		
5.3.1 如何设置和删除断点	119		

7.4.3 MCS-51 单片机中断系统的 应用举例	167	第 9 章 51 单片机与 D/A、 A/D 转换器的接口	217
习题	173		
第 8 章 MCS-51 单片机的常用接口	175	9.1 D/A 转换器与 51 单片机的接口	217
8.1 MCS-51 单片机的最小系统	175	9.1.1 D/A 转换器概述	217
8.1.1 8051/8751 的最小系统	175	9.1.2 典型的 D/A 转换器 芯片 DAC0832	220
8.1.2 8031 的最小系统	176	9.1.3 DAC0832 与 51 单片机的 接口与应用	222
8.2 存储器扩展	176	9.2 A/D 转换器与 51 单片机的接口	228
8.2.1 半导体存储器概述	176	9.2.1 A/D 转换器概述	228
8.2.2 存储器扩展的一般方法	179	9.2.2 典型的 A/D 转换器 芯片 ADC0808/0809	230
8.2.3 程序存储器的扩展	182	习题	237
8.2.4 数据存储器扩展	184	第 10 章 MCS-51 单片机的其他接口	239
8.3 输入/输出扩展	185	10.1 LCD1602 与 51 单片机的接口	239
8.3.1 简单 I/O 接口扩展	185	10.1.1 LCD1602 概述	239
8.3.2 可编程 I/O 接口 扩展(8255A)	187	10.1.2 LCD1602 的内部结构	240
8.4 LED 显示器与 51 单片机接口	194	10.1.3 HD44780 的指令格式与 指令功能	242
8.4.1 LED 显示器的基本结构与 原理	194	10.1.4 LCD1602 的编程与接口	244
8.4.2 LED 数码管显示器使用的 主要问题	196	10.2 I ² C 总线芯片与 51 单片机接口	248
8.4.3 LED 显示器与单片机的 接口	198	10.2.1 I ² C 总线简介	248
8.5 键盘与 51 单片机的接口	202	10.2.2 I ² C 总线 EEPROM 芯片与 单片机的接口	251
8.5.1 键盘概述	202	10.3 日历时钟芯片 DS1302 与 51 单片机接口	267
8.5.2 独立式键盘与单片机的 接口	205	10.3.1 DS1302 简介	267
8.5.3 矩阵式键盘与单片机的 接口	207	10.3.2 DS1302 引脚的功能	268
8.6 行程开关、晶闸管、继电器与 51 单片机的接口	213	10.3.3 DS1302 的寄存器及 片内 RAM	268
8.6.1 行程开关、继电器与 51 单片机的接口	213	10.3.4 DS1302 与 51 单片机的 接口	270
8.6.2 晶闸管与 51 单片机接口	213	10.4 温度传感器 DS18B20 与 51 单片机的接口	280
8.6.3 继电器与 51 单片机的接口	214	10.4.1 DS18B20 简介	280
8.6.4 蜂鸣器与 51 单片机的接口	215	10.4.2 DS18B20 的外部结构	281
习题	215	10.4.3 DS18B20 的内部结构	281
		10.4.4 DS18B20 的温度转换 过程	283

10.4.5 DS18B20 与 51 单片机的 接口	285	11.3 多路数字电压表的设计	336
10.5 串行 A/D、D/A 与 51 单片机的 接口	293	11.3.1 多路数字电压表的 功能要求	336
10.5.1 串行 A/D 芯片 MAX1241 与 51 单片机的接口	293	11.3.2 多路数字电压表的 总体设计	336
10.5.2 串行 D/A 芯片 MAX517 与 51 单片机的接口	301	11.3.3 多路数字电压表的 硬件电路	336
习题	306	11.3.4 多路数字电压表的 软件程序	337
第 11 章 单片机应用系统设计及举例	307	11.4 多点温度测量系统的设计	345
11.1 单片机应用系统的开发过程	307	11.4.1 多点温度测量系统的 功能要求	345
11.1.1 应具备的知识和能力	307	11.4.2 多点温度测量系统的 总体设计	345
11.1.2 单片机应用系统开发的 基本过程	308	11.4.3 多点温度测量系统的 硬件电路	346
11.1.3 单片机应用系统的 硬件设计	309	11.4.4 多点温度测量系统的 软件程序	346
11.1.4 单片机应用系统的 软件设计	311	习题	361
11.2 电子时钟的设计	312	附录 A MCS-51 系列单片机指令表	363
11.2.1 电子时钟的功能要求	312	附录 B C51 运算符优先级和结合规则	368
11.2.2 总体方案设计	312	附录 C C51 的库函数	370
11.2.3 软件计时数码管显示时钟的 硬件电路	313	附录 D 单片机技术及嵌入式系统的 网络资源	381
11.2.4 软件计时数码管显示时钟的 软件程序	314	参考文献	382
11.2.5 硬件定时液晶显示时钟的 硬件电路	321		
11.2.6 硬件定时液晶显示时钟的 软件程序	322		

第 1 章 基础知识

1.1 信息在计算机中的表示

计算机是能够对输入的信息进行加工处理、存储并能按要求输出结果的电子设备，又称为电脑或信息处理机。现在使用的计算机是按照冯·诺依曼(Von Neumann)的存储程序原理工作的，内部按二进制数进行运算。任何信息，不管是数字还是字符，在计算机中都是以二进制编码形式进行表示和处理的。在学习计算机内部信息的处理、表示之前，先介绍一下计算机中信息的表示。

1.1.1 数在计算机内的表示

计算机中的数通常有两种：无符号数和有符号数。这两种数在计算机中的表示是不一样的。

无符号数由于不带符号，表示时比较简单，可以直接用它对应的二进制形式表示。例如，假设机器字长为 8 位，则 123 表示成 01111011B。

有符号数带有正负号。数学上用正负号来表示数的正负。由于计算机只能识别二进制符号，不能识别正负号，因此计算机中只能将正、负号数字化，用二进制数字表示。通常，在计算机中表示有符号数时，在数的前面加一位，作为符号位。正数表示为 0，负数表示为 1，其余的位用于表示数的大小。这种连同符号位在一起的数，称为机器数，它的数值称为机器数的真值。机器数的表示如图 1.1 所示。

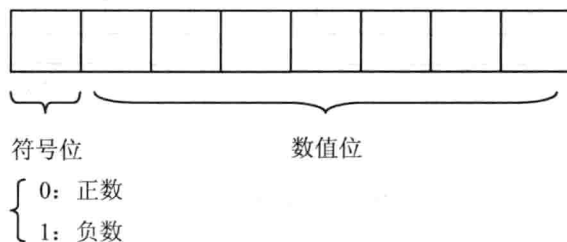


图 1.1 机器数的表示

在计算机的发展过程中，机器数先后有 3 种表示法：原码、反码和补码。

1. 原码

原码表示时，最高位为符号位，正数用 0 表示，负数用 1 表示，其余的位用于表示数的绝对值。正数的符号位为 0，因而正数的表示与它对应的无符号数的表示相同，负数则不是。原码的表示如图 1.2 所示。

原码表示时，由于最高位用作符号位，剩下的位作为数的绝对值位。对于一个 n 位的二进制数，其原码表示范围为 $-(2^{n-1}-1) \sim +(2^{n-1}-1)$ 。例如，如果用 8 位二进制表示原码，

则数的范围为 $-127 \sim +127$ 。

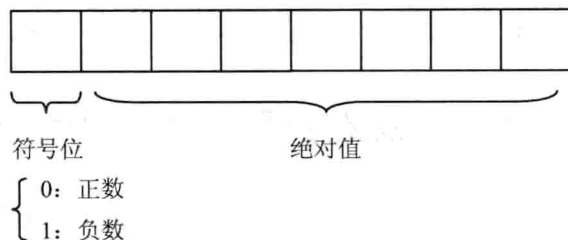


图 1.2 原码的表示

以原码表示时,对于 -0 和 $+0$ 的编码是不一样的。假设机器字长为 8 位, -0 的编码为 10000000B , $+0$ 的编码为 00000000B 。

【例 1.1】求 $+67$ 、 -25 的原码(机器字长 8 位)。

因为:

$$\begin{aligned} | +67 | &= 67 = 1000011\text{B} \\ | -25 | &= 25 = 11001\text{B} \end{aligned}$$

所以:

$$\begin{aligned} [+67]_{\text{原}} &= 01000011\text{B} \\ [-25]_{\text{原}} &= 10011001\text{B} \end{aligned}$$

2. 反码

以反码表示时,最高位为符号位,正数用 0 表示,负数用 1 表示。正数的反码与原码相同,而负数的反码可在原码的基础之上,符号位不变,其余位取反得到。反码的表示如图 1.3 所示。

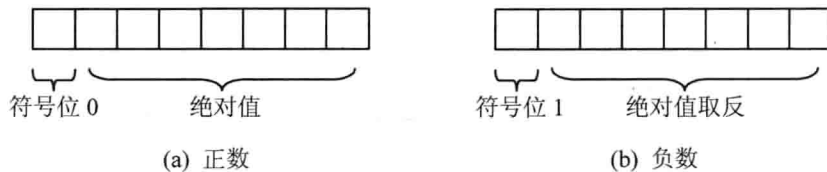


图 1.3 反码的表示

反码数的表示范围与原码相同,对于一个 n 位的二进制,它的反码表示范围为 $-(2^{n-1}-1) \sim +(2^{n-1}-1)$,对于 0,假设机器字长为 8 位, -0 的反码为 11111111B , $+0$ 的反码为 00000000B 。

【例 1.2】求 $+67$ 、 -25 的反码(机器字长 8 位)。

因为:

$$\begin{aligned} [+67]_{\text{原}} &= 01000011\text{B} \\ [-25]_{\text{原}} &= 10011001\text{B} \end{aligned}$$

所以:

$$\begin{aligned} [+67]_{\text{反}} &= 01000011\text{B} \\ [-25]_{\text{反}} &= 11100110\text{B} \end{aligned}$$

3. 补码

以补码表示时，最高位为符号位，正数用 0 表示，负数用 1 表示。正数的补码与原码相同，而负数的补码可在原码的基础之上，符号位不变，其余位取反，末位加 1 得到。对于一个负数 X ，其补码也可用 $2^n - |X|$ 得到，其中 n 为计算机字长。补码的表示如图 1.4 所示。

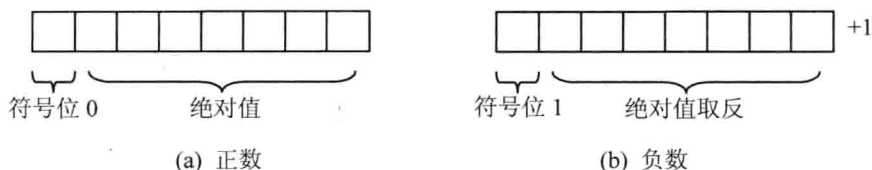


图 1.4 补码的表示

【例 1.3】 求 +67、-25 的补码(机器字长 8 位)。

因为：

$$[+67]_{\text{原}} = 01000011\text{B}$$

$$[-25]_{\text{原}} = 10011001\text{B}$$

所以：

$$[+67]_{\text{补}} = 01000011\text{B}$$

$$[-25]_{\text{补}} = 11100111\text{B}$$

另外，对于计算补码，也可用一种求补运算方法求得。

求补运算：一个二进制数，符号位和数值位一起取反，末位加 1。

求补运算具有以下特点。

对于一个数 X ：

$$[X]_{\text{补}} \xrightarrow{\text{求补}} [-X]_{\text{补}}$$

所以，已知正数的补码，则可通过求补运算求得对应负数的补码，已知负数的补码，相应也可通过求补运算求得对应正数的补码，也就是说，在用补码表示时，求补运算可得到数的相反数。

【例 1.4】 已知 +25 的补码为 00011001B，用求补运算求 -25 的补码。

$$\text{因为：} \quad [25]_{\text{补}} \xrightarrow{\text{求补}} [-25]_{\text{补}}$$

所以：

$$[-25]_{\text{补}} = 11100110 + 1 = 11100111\text{B}$$

对于一个 n 位的二进制数，其补码的表示范围为 $-(2^{n-1}) \sim +(2^{n-1}-1)$ 。

以补码表示时，对于 -0 和 +0 来讲，其补码是相同的，假设机器字长为 8 位，则 0 的补码为 00000000B。

4. 补码的加减运算

补码的加、减法运算规则如下：

$$[X+Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}}$$

$$[X-Y]_{\text{补}} = [X]_{\text{补}} - [Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}} = [X]_{\text{补}} + \{[Y]_{\text{补}}\}_{\text{求补}}$$

即：补码表示时，求两个数之和 $[X + Y]_{\text{补}}$ ，直接用两个数相加 $[X]_{\text{补}} + [Y]_{\text{补}}$ ；求两个数之差 $[X - Y]_{\text{补}}$ ，可以直接用两个数相减 $[X]_{\text{补}} - [Y]_{\text{补}}$ ，也可以先对减数 $[Y]_{\text{补}}$ 求补运算，然后再与被减数 $[X]_{\text{补}}$ 相加。也就是说，减法运算可通过加法和求补运算来处理。

【例 1.5】 假设计算机字长为 8 位，完成下列补码运算。

① $(+25) + (+32)$

$$\begin{array}{r} \text{因为：} \quad [+25]_{\text{补}} = 00011001\text{B} \quad [+32]_{\text{补}} = 00100000\text{B} \\ \qquad \qquad \qquad [+25]_{\text{补}} = 00011001 \\ \qquad \qquad \qquad + \quad [+32]_{\text{补}} = 00100000 \\ \hline \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad 00111001 \end{array}$$

所以： $[(+25) + (+32)]_{\text{补}} = [+25]_{\text{补}} + [+32]_{\text{补}} = 00111001\text{B} = [+57]_{\text{补}}$

② $(+25) + (-32)$

$$\begin{array}{r} \text{因为：} \quad [+25]_{\text{补}} = 00011001\text{B} \quad [-32]_{\text{补}} = 11100000\text{B} \\ \qquad \qquad \qquad [+25]_{\text{补}} = 00011001 \\ \qquad \qquad \qquad + \quad [-32]_{\text{补}} = 11100000 \\ \hline \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad 11111001 \end{array}$$

所以： $[(+25) + (-32)]_{\text{补}} = [+25]_{\text{补}} + [-32]_{\text{补}} = 11111001\text{B} = [-7]_{\text{补}}$

③ $(+25) - (+32)$

$$\begin{array}{r} \text{因为：} \quad [+25]_{\text{补}} = 00011001\text{B} \quad [+32]_{\text{补}} = 00100000\text{B} \\ \qquad \qquad \qquad \{ [+32]_{\text{补}} \}_{\text{求补}} = 11011111 + 1 = 11100000\text{B} \\ \qquad \qquad \qquad [+25]_{\text{补}} = 00011001 \qquad \qquad \qquad [+25]_{\text{补}} = 00011001 \\ \qquad \qquad \qquad - \quad [+32]_{\text{补}} = 00100000 \qquad \qquad \qquad + \quad [-32]_{\text{补}} = 11100000 \\ \hline \text{借位 1 自动丢失} \longrightarrow 1 \quad 11111001 \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad 11111001 \end{array}$$

所以： $[(+25) - (+32)]_{\text{补}} = [+25]_{\text{补}} + [-32]_{\text{补}} = [+25]_{\text{补}} + \{ [+32]_{\text{补}} \}_{\text{求补}} = 11111001\text{B} = [-7]_{\text{补}}$

④ $(+25) - (-32)$

$$\begin{array}{r} \text{因为：} \quad [+25]_{\text{补}} = 00011001\text{B} \quad [-32]_{\text{补}} = 11100000\text{B} \\ \qquad \qquad \qquad \{ [-32]_{\text{补}} \}_{\text{求补}} = 00011111 + 1 = 00100000\text{B} \\ \qquad \qquad \qquad [+25]_{\text{补}} = 00011001 \qquad \qquad \qquad [+25]_{\text{补}} = 00011001 \\ \qquad \qquad \qquad - \quad [-32]_{\text{补}} = 11100000 \qquad \qquad \qquad + \quad [+32]_{\text{补}} = 00100000 \\ \hline \text{借位 1 自动丢失} \longrightarrow 1 \quad 00111001 \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad 00111001 \end{array}$$

所以： $[(+25) - (-32)]_{\text{补}} = [+25]_{\text{补}} + [-32]_{\text{补}} = [+25]_{\text{补}} + \{ [-32]_{\text{补}} \}_{\text{求补}} = 00111001\text{B} = [+57]_{\text{补}}$

从以上可以看出，通过补码进行加减运算非常方便，而且能把减法转换成加法，这样就可进一步简化计算机中运算器的线路设计，所以，现在的计算机中，有符号数都用补码来表示。

5. 十进制数的表示

计算机内部对信息是按二进制方式进行处理的，但我们生活中习惯于使用十进制。为了处理方便，在计算机中，对十进制数也提供了十进制编码形式。

十进制编码又称为 BCD 码，分为压缩 BCD 码和非压缩 BCD 码。压缩 BCD 码又称

为 8421 码, 它用四位二进制编码来表示一位十进制符号。十进制数符号有 0~9 十个, 编码情况如表 1.1 所示。

表 1.1 压缩 BCD 编码

十进制符号	压缩 BCD 编码	十进制符号	压缩 BCD 编码
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

用压缩 BCD 码表示十进制数, 只要把每个十进制符号用对应的四位二进制编码代替即可。例如, 十进制数 24 的压缩 BCD 码为 0010 0100。

非压缩 BCD 码是用八位二进制编码来表示一位十进制符号, 其中低四位二进制编码与压缩 BCD 码相同, 高四位任取。例如, 下面介绍的数字符号的 ASCII 码就是一种非压缩的 BCD 码。用非压缩 BCD 码表示十进制数, 一位十进制符号须用八位二进制数表示。例如, 十进制数 24 的非压缩 BCD 码为 0011 0010 0011 0100。

1.1.2 字符在计算机内的表示

在计算机信息处理中, 除了处理数值数据外, 还涉及大量的字符数据。例如, 从键盘上输入的信息或打印输出的信息都是以字符方式输入/输出的, 字符数据包括字母、数字、专用字符及一些控制字符等, 这些字符在计算机中也是用二进制编码表示的。现在的计算机中, 字符数据的编码通常采用的是美国信息交换标准代码 ASCII(American Standard Code for Information Interchange)。基本 ASCII 码标准定义了 128 个字符, 用七位二进制数来编码, 包括英文 26 个大写字母, 26 个小写字母、10 个数字符号 0~9, 还有一些专用符号(如“:”、“!”、“%”)及控制符号(如换行、换页、回车等)。常用字符的 ASCII 码如表 1.2 所示。计算机中一般以字节为单位, 而 8 位二进制表示一个字节, 字符 ASCII 码通常放于低 7 位, 高位一般补 0, 在通信时, 最高位常用作奇偶校验位。

表 1.2 常用字符的 ASCII 码(十六进制表示)

字符	ASCII	字符	ASCII	字符	ASCII	字符	ASCII	字符	ASCII
NUL	00	.	2F	C	43	W	57	k	6B
BEL	07	0	30	D	44	X	58	l	6C
LF	0A	1	31	E	45	Y	59	m	6D
FF	0C	2	32	F	46	Z	5A	n	6E
CR	0D	3	33	G	47	[5B	o	6F
SP	20	4	34	H	48	\	5C	p	70
!	21	5	35	I	49]	5D	q	71

续表

字符	ASCII	字符	ASCII	字符	ASCII	字符	ASCII	字符	ASCII
"	22	6	36	J	4A	↑	5E	r	72
#	23	7	37	K	4B	'	5F	s	73
\$	24	8	38	L	4C	←	60	t	74
%	25	9	39	M	4D	a	61	u	75
&	26	:	3A	N	4E	b	62	v	76
'	27	;	3B	O	4F	c	63	w	77
(28	<	3C	P	50	d	64	x	78
)	29	=	3D	Q	51	e	65	y	79
*	2A	>	3E	R	52	f	66	z	7A
+	2B	?	3F	S	53	g	67	{	7B
,	2C	@	40	T	54	h	68		7C
-	2D	A	41	U	55	i	69	}	7D
/	2E	B	42	V	56	j	6A	~	7E

1.2 微型计算机的基本结构和工作原理

根据冯·诺伊曼设计思想,现代的计算机由运算器、控制器、存储器、输入设备和输出设备五大部分组成。微型计算机是计算机发展到一定阶段的产物,由于大规模集成电路技术的发展,使得制造商能够把运算器和控制器集成在一块集成电路芯片内,通常把集成运算器和控制器的这一块集成电路称为中央处理器或微处理器,简称 CPU。微型计算机(Micro Computer)是指以中央处理器为核心,配上存储器、输入/输出接口电路等所组成的计算机。微型计算机系统(Micro Computer System)是指以微型计算机为中心,配以相应的外围设备、电源和辅助电路以及指挥计算机工作的系统软件所构成的系统。

1.2.1 微型计算机的发展

微型计算机的出现,是计算机技术发展史上的一个新的里程碑,为计算机技术的发展和普及开辟了崭新的途径。我们日常生活中使用的大部分都是微型计算机。

说到微型计算机,必须要提到 Intel 公司。Intel 公司是一个生产微处理器 CPU 的公司,1971 年诞生的第一台微型计算机就是用 Intel 公司生产的 Intel 4004 为处理器的。Intel 公司在整个微型计算机的发展中都起着非常重要的主导作用,正是由于 Intel 公司不断开发新型、功能强大的微处理器,才推动了微型计算机不断向前发展。

微型计算机按 CPU 字长和功能,一般分为以下几代。

- 第一代(1971—1973): 4 位和低档 8 位微处理器,代表产品为 4004→4040→8008 系列。
- 第二代(1974—1977): 中高档 8 位微处理器,代表产品为 Z80、I8085、MC6800 系列,以及 Apple-II 微机。