

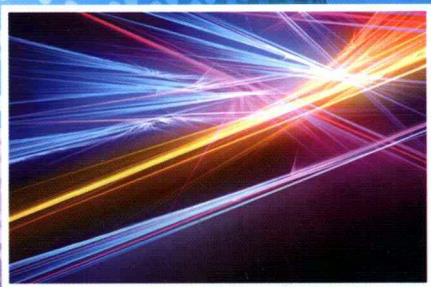


普通高等教育“十二五”规划教材

◎ 电子信息科学与工程类专业 规划教材

单片机原理、接口 及应用系统设计

◎ 谢维成 杨加国 主编



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

电子信息科学与工程类专业规划教材

单片机原理、接口 及应用系统设计

谢维成 杨加国 主编

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

MCS-51 单片机是学习单片机技术较好的平台,同时也是开发单片机应用系统的 MCU 的一个重要系列。本书以实用为宗旨,用丰富的实例讲解 MCS-51 单片机原理和软硬件开发技术,并采用对比的方法,同一功能分别用单片机汇编语言和单片机 C 语言来实现,特别突出应用系统设计方法,并提供单片机应用系统设计实训参考方案。

全书共分 11 章,第 1~4 章介绍单片微机系统的原理,包括计算机基础知识及微处理器、MCS-51 单片机原理与结构、MCS-51 单片机指令系统和 MCS-51 单片机程序设计;第 5~7 章,用实例介绍 MCS-51 单片机接口技术及应用,包括 MCS-51 单片机常用接口、MCS-51 单片机与 D/A、A/D 转换器的接口、MCS-51 单片机的其他接口;第 8~11 章介绍单片机应用系统设计,包括单片机应用系统设计及举例、Keil μ Vision IDE 集成环境的使用、Proteus 软件的使用、单片机应用系统设计实训方案;附录提供了 MCS-51 系列单片机指令表、C51 库函数及单片机技术相关的网络资源。

本书适合各类本科和专科院校及培训机构作为“单片机原理与应用”等课程的教材,特别适合学习单片机应用系统开发的读者,也可供信息、测控、电气、自动化、计算机、机电等各类技术人员和计算机爱好者学习参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

单片机原理、接口及应用系统设计/ 谢维成, 杨加国主编. —北京: 电子工业出版社, 2011.11
电子信息科学与工程类专业规划教材
ISBN 978-7-121-14970-2

I. ①单… II. ①谢… ②杨… III. ①单片微型计算机—基础理论—高等学校—教材②单片微型计算机—接口—高等学校—教材③单片微型计算机—系统设计—高等学校—教材 IV. ①TP368.1

中国版本图书馆 CIP 数据核字(2011)第 223925 号

责任编辑: 凌 毅 特约编辑: 张 莉

印 刷: 北京市海淀区四季青印刷厂

装 订: 三河市鹏成印业有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 19.5 字数: 530 千字

印 次: 2011 年 11 月第 1 次印刷

印 数: 3000 册 定价: 36.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

MCS-51 系列单片机的应用很广泛，是学习单片机技术较好的系统平台，同时也是开发单片机应用系统的一个重要 CPU 系列。目前，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，但汇编语言学习起来会比较困难。在实际的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 语言不仅学习方便，而且同汇编语言一样也能够对单片机的资源进行访问，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，在选用教材时发现存在几个方面的问题：

第一，单片机原理与应用（含单片机 C 语言程序设计）的教材很多，而兼顾汇编语言和 C 语言的教材很少，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。

第三，在减少学分的大背景下，有的学校在未开设“微型计算机原理”课程的情况下直接开设“单片机原理及应用”课程，目前的大部分单片机技术教材显得入门困难。

而我们需要一本在不开设“微型计算机原理”课程的情况下，直接学习单片机技术的教材。可以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，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。基于此，我们在 2005 年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”，并于 2008 年获得四川省高等教育教学成果奖三等奖。我们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，我们发现学生要想熟练掌握 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 C51 程序设计。因此我们在课堂上讲解单片机原理的同时兼顾汇编语言和 C 语言两个方面，对比讲解，效果明显，避免直到进入实验室或开发实践阶段时才讲授单片机 C 语言程序设计及开发环境，为开设综合实验和创新性实验及单片机应用系统设计奠定一定的基础。

2006 年我们在清华大学出版社出版了《单片机原理与应用及 C51 程序设计》一书，以其新的编写思路、鲜明的应用性特色，受到了广大教师与学生的欢迎，已经重印 9 次。2009 年对第 1 版进行了修订和补充，出版了《单片机原理与应用及 C51 程序设计（第 2 版）》，更加适合各院校使用，已经重印 8 次。编者认真听取了广大师生的意见，为解决未开设“微型计算机原理”课程直接开设“单片机原理及应用”课程的需求，均衡内容与篇幅，我们保留了《单片机原理与应用及 C51 程序设计》的核心思路，增加了“微型计算机原理”的基础知识，并重新组织了内容，分成单片机原理、接口及应用系统设计三部分讲解，更加突出应用系统设计方法，适当加入了 Proteus ISIS 仿真的内容，特别增加了单片机应用系统设计实训参考方案的内容。

在本书的实例中，相同的功能分别用汇编语言和 C 语言来实现，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语言并认识另一种语言。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. 本书特点

本书以实用为宗旨，用众多的实例讲解了 MCS-51 单片机原理和软硬件开发技术，针对同

一功能，同时提供单片机汇编源程序和单片机 C 语言源程序，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，有利于初学者迅速掌握单片机技术，并且可以在未学“微型计算机原理”的情况下直接学习“单片机原理及应用”。

本书图文并茂，实用性强，为便于读者自学和练习，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，特别适合于打算学习单片机应用系统开发技术的读者，同时可供信息、测控、计算机、机电、自动化等各类技术人员和计算机爱好者学习参考。

2. 本书内容

本书共分 11 章，从内容上可分 4 部分，具体内容如下。

第一部分：单片微机系统原理

第 1 章，计算机基础知识及微处理器，主要介绍单片微机系统必备的基础知识、微处理器和单片机。

第 2 章，MCS-51 单片机原理与结构，详细介绍 MCS-51 单片机的工作原理。

第 3 章，MCS-51 单片机指令系统，介绍包括寻址方式、MCS-51 单片机指令系统等汇编源程序设计的相关技术细节。

第 4 章，MCS-51 单片机程序设计，主要介绍 MCS-51 单片机汇编程序设计和 C 语言程序设计，并列举了大量实例及详细代码，在应用的时候可以直接修改使用。

第二部分：单片机接口技术及应用

第 5 章，MCS-51 单片机常用接口。

第 6 章，MCS-51 单片机与 D/A、A/D 转换器的接口。

第 7 章，MCS-51 单片机的其他接口。

第三部分：单片机应用系统设计

第 8 章，单片机应用系统设计及举例，用 3 个简单的单片机应用系统设计为例来讲解单片机应用系统的设计技术。

第 9 章，Keil μ Vision IDE 集成环境的使用。

第 10 章，Proteus 软件的使用。

第 11 章，单片机应用系统设计实训。

第四部分：与单片机技术相关的附录

附录中提供了 MCS-51 系列单片机指令表和 C51 的库函数，以及与单片机相关的资源网站列表，以使读者找到更广阔的学习园地。

3. 如何使用本书

对于未学习“微型计算机原理”课程的读者或者 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，以了解微机的工作原理和 MCS-51 单片机技术的基本知识，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，学完第 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 50~60 学时，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减少。课程学完后，可安排相应的应用系统设计实训或课程设计，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 单片机简称为“单片机”，书中采用了 Keil C51 和 Proteus ISIS 软件界面，读者在学习过程中也可以采用 Keil C51 和 Proteus ISIS 的最新版本，或者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。若需要单片机应用系统设计实训母板的读者，可联系 scxweicheng@yahoo.com.cn。

4. 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，关键是让读者迅速找到适合自己的学习方法，在第一时间使读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，然后由浅入深，逐步进入单片机应用系统开发领域。

为此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可从华信教育资源网 www.hxedu.com.cn 免费下载，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中选择一个单片机应用系统实训项目进行开发练习，以此作为真正的单片机应用系统开发的起步。

5. 致谢

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

本书第 1、9 章由蒋文波编写，第 2、3、5、6 章由杨加国编写，第 4、7、8、11 章由谢维成编写，第 10 章和附录由杨显富编写，最后由谢维成和杨加国统稿完成。另外，王胜、郑海春、王孝平、陈永强、赵华颖参与了本书部分章节图形的绘制以及程序的调试工作，在此一并表示感谢。同时感谢参考文献中提到的作者，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 计算机基础知识及微处理器	1
1.1 计算机中的信息及表示	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	6
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	7
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 微型计算机工作过程	17
1.3 单片机、51 单片机及其系列	20
1.3.1 单片机的基本概念	20
1.3.2 单片机的主要特点	20
1.3.3 单片机的发展及其主要品种	21
1.3.4 单片机的应用	25
习题	26
第 2 章 MCS-51 单片机原理与结构	27
2.1 MCS-51 单片机概述	27
2.1.1 MCS-51 单片机简介	27
2.1.2 MCS-51 单片机的基本组成	28
2.2 MCS-51 单片机的内部结构	28
2.2.1 MCS-51 单片机的中央处 理器	28
2.2.2 MCS-51 单片机的存储器	31
2.3 MCS-51 单片机的输入/输出接口	36
2.3.1 P0 口	36
2.3.2 P1 口	37
2.3.3 P2 口	38
2.3.4 P3 口	38
2.4 MCS-51 单片机定时/计数器	39
2.4.1 定时/计数器的主要特性	40
2.4.2 定时/计数器 T0、T1 的结构及 工作原理	40
2.4.3 定时/计数器的方式和控制寄 存器	41
2.4.4 定时/计数器的工作方式	42
2.5 MCS-51 单片机串行接口	44
2.5.1 通信的基本概念	44
2.5.2 MCS-51 单片机串行口的功能 与结构	45
2.5.3 串行口的工作方式	48
2.6 MCS-51 单片机中断系统	49
2.6.1 中断的基本概念	49
2.6.2 MCS-51 单片机的中断系统	50
2.7 MCS-51 单片机外部引脚及功能	54
2.7.1 输入/输出引脚	55
2.7.2 控制引脚	55
2.7.3 电源与晶振引脚	55
2.8 MCS-51 单片机工作方式与时序	56
2.8.1 MCS-51 单片机的工作方式	56
2.8.2 MCS-51 单片机的时序	58
习题	61
第 3 章 MCS-51 单片机指令系统	62
3.1 指令系统概述	62
3.1.1 指令格式	62
3.1.2 指令的字节数	63
3.1.3 MCS-51 单片机汇编指令常用 符号	64
3.2 MCS-51 单片机的寻址方式	64
3.2.1 常数寻址——立即寻址	64
3.2.2 寄存器数寻址——寄存器寻址	65
3.2.3 存储器数寻址	65
3.2.4 位寻址	66
3.2.5 指令寻址	66
3.3 MCS-51 单片机的指令系统	67
3.3.1 数据传送指令	67
3.3.2 算术运算指令	70
3.3.3 逻辑操作指令	73

3.3.4	控制转移指令	75	5.4.1	LED 显示器的基本结构与原理	136
3.3.5	位操作指令	80	5.4.2	LED 数码管显示器使用的主要问题	137
	习题	82	5.4.3	LED 显示器与单片机的接口	139
第 4 章	MCS-51 单片机程序设计	84	5.5	MCS-51 单片机与键盘的接口	142
4.1	MCS-51 单片机编程语言简介	84	5.5.1	键盘概述	142
4.1.1	单片机汇编语言的特点	84	5.5.2	独立式键盘与单片机的接口	144
4.1.2	单片机 C 语言的特点	85	5.5.3	矩阵式键盘与单片机的接口	145
4.2	MCS-51 单片机汇编语言常用伪指令	86		习题	150
4.3	MCS-51 单片机汇编程序设计	88	第 6 章	MCS-51 单片机与 D/A、A/D 转换器的接口	152
4.3.1	数据传送程序	88	6.1	MCS-51 单片机与 DAC 的接口	152
4.3.2	运算程序	89	6.1.1	D/A 转换器的基本原理	152
4.3.3	数据转换程序	92	6.1.2	D/A 转换器的性能指标	153
4.3.4	多分支转移(散转)程序	92	6.1.3	D/A 转换器的分类	154
4.3.5	延时程序	94	6.1.4	典型的 D/A 转换器芯片 DAC0832	155
4.4	C51 基本知识	94	6.1.5	DAC0832 与 MCS-51 单片机的接口与应用	157
4.4.1	C51 的数据类型	95	6.2	MCS-51 单片机与 ADC 的接口	162
4.4.2	C51 的变量与存储类型	97	6.2.1	A/D 转换器概述	162
4.4.3	绝对地址的访问	102	6.2.2	典型的 A/D 转换器芯片 ADC0808/0809	164
4.4.4	C51 中的函数	103		习题	169
4.5	MCS-51 单片机内部资源的编程	107	第 7 章	MCS-51 单片机的其他接口	170
4.5.1	并行口的编程与应用	107	7.1	LCD1602 与 MCS-51 单片机的接口	170
4.5.2	定时/计数器的初始化编程及应用	108	7.1.1	LCD1602 概述	170
4.5.3	串行口的编程及应用	112	7.1.2	LCD1602 的内部结构	171
4.5.4	MCS-51 单片机中断系统的应用	117	7.1.3	HD44780 的指令格式与指令功能	172
	习题	118	7.1.4	LCD1602 的编程与接口	174
第 5 章	MCS-51 单片机常用接口	120	7.2	I ² C 总线芯片与 MCS-51 单片机接口	178
5.1	MCS-51 单片机的最小系统	120	7.2.1	I ² C 总线简介	178
5.1.1	8051/8751 的最小系统	120	7.2.2	I ² C 总线 EEPROM 芯片与单片机的接口	180
5.1.2	8031 的最小系统	121	7.3	日历时芯芯片 DS1302 与 MCS-51 单片机接口	195
5.2	存储器扩展	121	7.3.1	DS1302 简介	195
5.2.1	半导体存储器概述	121	7.3.2	DS1302 引脚功能	195
5.2.2	存储器扩展的一般方法	124	7.3.3	DS1302 的寄存器及片内 RAM	195
5.2.3	程序存储器扩展	125			
5.2.4	数据存储器扩展	127			
5.3	输入/输出接口扩展	128			
5.3.1	简单 I/O 接口扩展	128			
5.3.2	可编程 I/O 接口扩展 (8255A)	130			
5.4	MCS-51 单片机与 LED 显示器接口	136			

7.3.4 DS1302 与单片机的接口	197	9.2 Keil μ Vision IDE 的使用方法	260
7.4 温度传感器 DS18B20 与 MCS-51 单片机的接口	206	9.2.1 项目文件的建立	260
7.4.1 DS18B20 简介	206	9.2.2 给项目添加程序文件	261
7.4.2 DS18B20 的外部结构	207	9.2.3 编译、连接项目, 形成目标文件	262
7.4.3 DS18B20 的内部结构	207	9.2.4 运行调试观察结果	262
7.4.4 DS18B20 的温度转换过程	209	9.2.5 仿真环境的设置	263
7.4.5 DS18B20 与 MCS-51 单片机的接口	210	9.3 Keil μ Vision IDE 的调试技巧	266
习题	218	9.3.1 如何设置和删除断点	266
第 8 章 单片机应用系统设计及举例	219	9.3.2 如何查看和修改寄存器的内容	266
8.1 单片机应用系统的开发过程	219	9.3.3 如何观察和修改变量	266
8.1.1 应具备的知识和能力	219	9.3.4 如何观察存储器区域	266
8.1.2 单片机应用系统开发的基本过程	220	习题	267
8.1.3 单片机应用系统的硬件设计 ..	221	第 10 章 Proteus 软件的使用	268
8.1.4 单片机应用系统的软件设计 ..	222	10.1 Proteus 概述	268
8.1.5 软、硬件仿真及开发工具的选择	223	10.1.1 Proteus 的进入	268
8.2 单片机电子时钟的设计	223	10.1.2 Proteus 的界面	269
8.2.1 单片机电子时钟的功能要求 ..	224	10.2 Proteus 的基本操作	271
8.2.2 总体方案设计	224	10.2.1 新建电路, 选择元件	271
8.2.3 软件计时数码管显示时钟硬件电路	224	10.2.2 放置元件, 调整元件	273
8.2.4 软件计时数码管显示时钟软件程序	225	10.2.3 连接导线	274
8.2.5 硬件定时液晶显示时钟硬件电路	233	10.2.4 给单片机加载程序	276
8.2.6 硬件定时液晶显示时钟软件程序	233	10.2.5 运行仿真看结果	277
8.3 多路数字电压表的设计	246	习题	278
8.3.1 多路数字电压表的功能要求 ..	246	第 11 章 单片机应用系统设计实训	279
8.3.2 多路数字电压表的总体设计 ..	246	11.1 单片机应用系统设计评分标准	279
8.3.3 多路数字电压表硬件电路	246	11.2 单片机应用系统设计任务书	280
8.3.4 多路数字电压表软件程序	247	11.3 单片机应用系统设计报告格式及要求	281
习题	254	11.4 单片机应用系统设计考核表	283
第 9 章 Keil μVision IDE 集成环境的使用	255	11.5 单片机应用系统设计参考母板电路 ..	284
9.1 Keil μ Vision IDE 简介	255	11.6 单片机应用系统设计实训参考题目 ..	285
9.1.1 Keil μ Vision IDE 的安装	255	习题	287
9.1.2 Keil μ Vision IDE 界面	255	附录 A MCS-51 系列单片机指令表	288
		A.1 数据传送类指令	288
		A.2 算术操作类指令	289
		A.3 逻辑操作类指令	290
		A.4 控制转移类指令	290
		A.5 位操作类指令	291
		附录 B C51 的库函数	292

B.1 寄存器库函数 REG×××.H..... 292
B.2 字符函数 CTYPE.H..... 292
B.3 一般输入/输出函数 STDIO.H..... 293
B.4 内部函数 INTRINS.H..... 295
B.5 标准函数 STDLIB.H..... 296
B.6 字符串函数 STRING.H..... 297
B.7 数学函数 MATH.H..... 299
B.8 绝对地址访问函数 ABSACC.H..... 300

**附录 C 单片机技术及嵌入式系统的网
络资源**..... 301
C.1 单片机技术及嵌入式系统的常见
网站..... 301
C.2 单片机技术及嵌入式系统的官方
网站..... 301
参考文献..... 302

第 1 章 计算机基础知识及微处理器

主要内容:

作为本书的第 1 章, 主要对后面单片机学习中所需要的基本知识进行介绍。通过本章学习, 使读者了解信息在计算机内部的表示; 微型计算机的基础知识、基本结构和工作原理; 单片机的基本概念、特点和发展过程。

学习重点:

- ◆ 有符号数在计算机中的补码表示及运算方法
- ◆ 微型计算机的基本结构和工作过程
- ◆ 单片机的概念与特点

1.1 计算机中的信息及表示

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, 又称电脑或信息处理机。计算机中处理的信息, 主要有数值信息和非数值信息两大类。数值信息是指日常生活中接触到的数字类数据, 主要用来表示数量的多少, 可以比较大小; 非数值信息有多种, 有用来表示文字信息的字符数据, 也有用来表示图形、图像和声音等其他信息的数据。不同的信息在计算机中的表示形式不一样。从计算机内部角度来说, 由于计算机内部只能识别二进制数, 因此所有信息在计算机内都通过二进制编码表示。

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。两种数在计算机中的表示是不一样的。

1. 无符号数在计算机内的表示

无符号数不带符号, 表示时比较简单, 在计算机中一般直接用二进制数的形式表示, 位数不足时前面加 0 补充。对一个 n 位二进制数, 它能表示的无符号数的范围是 $0 \sim 2^n - 1$ 。例如, 假设机器字长 8 位, 无符号数 156 在计算机中表示是 10011100B, 45 在计算机中表示是 00101101B。

2. 有符号数在计算机内的表示

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, 用二进制符号表示。在计算机中表示有符号数时, 一般用二进制数的最高位来表示符号, 正数表示为 0, 负数表示为 1, 称为符号位; 其余位用来表示有符号数的数值大小, 称为数值位。通常, 把一个数及其符号位在计算机中的二进制数的表示形式称为“机器数”。机器数所表示的值称为该机器数的“真值”。机器数的表示如图 1.1 所示。

机器数通常有 3 种表示方法: 原码表示法、反码表示法和补码表示法。为了运算方便, 计算机中通常用补码表示。为了研究补码表示法, 首先了解原码表示法和反码表示法。

(1) 原码

原码表示方法如下: 最高位为符号位, 用 0 表示正数, 用 1 表示负数, 数值位用数的绝对

值表示，数值位如位数不足前面加 0 填充。由于正数的符号位为 0，因而正数的原码表示与相应的无符号数的表示相同。原码的表示如图 1.2 所示。

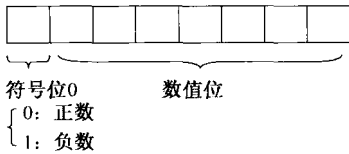


图 1.1 机器数的表示

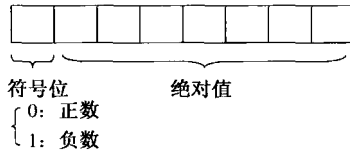


图 1.2 原码的表示

【例 1-1】求+78、-23 的原码（设机器字长 8 位）。

因为

$$\begin{aligned} | +78 | &= 78 = 1001110B \\ | -23 | &= 23 = 10111B \end{aligned}$$

所以

$$\begin{aligned} [+78]_{原} &= 01001110B \\ [-23]_{原} &= 10010111B \end{aligned}$$

原码表示时，如果机器字长为 n 位二进制数，其原码表示的有符号数范围为 $-(2^{n-1}-1) \sim +(2^{n-1}-1)$ 。例如，如果机器字长为 8 位二进制数，则表示的有符号数范围为 $-127 \sim +127$ 。

另外，“0”的原码表示有两个，-0 和+0 的编码不一样。假设机器字长为 8 位，-0 的编码为 10000000B，+0 的编码为 00000000B。

原码表示简单直观，且与真值的转换很方便，但不便于在计算机中进行加减运算，运算时符号位与数值位必须分开处理。两数相加时，必须先判断两个数的符号是否相同。如果相同，则符号位不变，对数值位相加得结果数值位；如果符号位不同，则应将数值位相减，数值位相减时，必须比较两数的数值位大小，再由大数减小数得结果的数值位，而结果的符号位要和数值位大的数的符号位一致。两数相减时情况类似。按上述运算方法设计的运算电路很复杂。为运算方便，后来计算机中引入了反码表示和补码表示。

(2) 反码

反码是在原码的基础上发展而来的，反码表示方法如下：最高位为符号位，用 0 表示正数，用 1 表示负数，对于数值位，正数的反码数值位与原码相同，而负数的反码数值位由原码的数值位取反得到。反码的表示如图 1.3 所示。

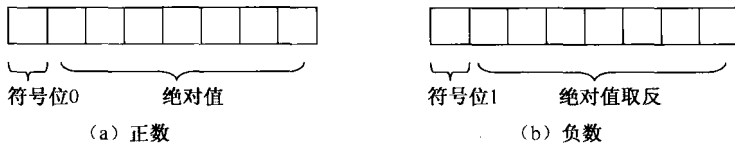


图 1.3 反码的表示

【例 1-2】求+78、-23 的反码（设机器字长 8 位）。

因为

$$\begin{aligned} [+78]_{原} &= 01001110B \\ [-23]_{原} &= 10010111B \end{aligned}$$

所以

$$[+78]_{反} = 01001110B$$

$$[-23]_{\text{反}}=11101000\text{B}$$

反码的表示范围与原码相同，如果机器字长为 n 位二进制数，则反码表示的有符号数范围为 $-(2^{n-1}-1) \sim +(2^{n-1}-1)$ 。例如，如果机器字长为 8 位二进制数，则表示的有符号数范围为 $-127 \sim +127$ 。

另外，“0”的反码表示也有两个。假设机器字长为 8 位， -0 的反码编码为 11111111B ， $+0$ 的反码编码为 00000000B 。

反码表示时，数的运算也不方便，也存在与原码相同的问题，因而，反码在计算机中用得很少，很快就被补码表示方法所替代。

(3) 补码

补码表示时，数的加减运算非常简单、方便，因而现在的计算机有符号数都用补码表示。补码表示如下：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、反码相同，而负数的补码可在反码的基础之上，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。

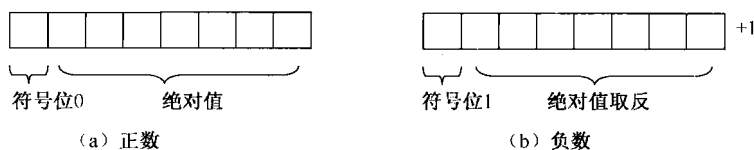


图 1.4 补码的表示

【例 1-3】 求 $+78$ 、 -23 的补码（设机器字长 8 位）。

因为

$$[+78]_{\text{反}}=01001110\text{B}$$

$$[-23]_{\text{反}}=11101000\text{B}$$

所以

$$[+78]_{\text{补}}=01001110\text{B}$$

$$[-23]_{\text{补}}=11101000\text{B}+1=11101001\text{B}$$

补码表示时，这里再介绍一种比较重要的运算：求补运算。

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。加 1 时可能向前面依次产生进位，最高位产生的进位自然丢失。

求补运算具有以下特点：对于一个任意的数 X

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

那么，在计算机中，如果已知一个正数的补码，则可通过求补运算求得对应负数的补码；如果已知一个负数的补码，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。由于正数的补码简单，很容易计算，因而它给我们提供了另外一种计算负数补码的方法，先计算出正数的补码，再用求补运算。另外，负数的补码转换成真值不方便，我们也可以通过先计算出相应的正数来得到。

【例 1-4】 用求补运算计算 -25 的补码（设机器字长 8 位）。

因为

$$[+25]_{\text{补}}=00010111$$

所以

$$[-25]_{\text{补}}=[+25]_{\text{补}}_{\text{求补}}=11101000+1=11101001\text{B}$$

补码表示时，如果机器字长为 n 位二进制数，则补码表示的有符号数范围为 $-(2^{n-1}-1) \sim +(2^{n-1}-1)$ 。例如，如果机器字长为 8 位二进制数，则表示的有符号数范围为 $-128 \sim +127$ 。 -0 和

+0 的补码是相同的, 假设机器字长为 8 位, 则 0 的补码为 00000000B。

(4) 补码的加减运算

在现在的计算机中, 有符号数的表示都用补码表示, 用补码表示时运算简单。

补码的加、减法运算规则如下:

$$[X+Y]_{补}=[X]_{补}+[Y]_{补}$$

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

即: 求两个数之和的补码, 直接用两个数的补码相加; 求两个数之差的补码, 用被减数的补码加减数的相反数的补码 $[-Y]_{补}$, 对于 $[-Y]_{补}$ 用 $[Y]_{补}$ 求补运算就可以得到, 也就是说, 减法运算可通过加法和求补运算来处理, 下面通过例子来看看补码的加减运算。

【例 1-5】 假设计算机字长为 8 位, 完成下列补码运算。

(1) $(+78)+(+23)$

因为 $[+78]_{补}=01001110B$, $[+23]_{补}=00010111B$

$$\begin{array}{r} [+ 7 8]_{补} = 0 1 0 0 1 1 1 0 \\ + [+ 2 3]_{补} = 0 0 0 1 0 1 1 1 \\ \hline 0 1 1 0 0 1 0 1 \end{array}$$

所以 $[(+78)+(+23)]_{补}=[+78]_{补}+[+23]_{补}=01100101B=[+101]_{补}$

(2) $(+78)+(-23)$

因为 $[+78]_{补}=01001110B$, $[-23]_{补}=11101001B$

$$\begin{array}{r} [+ 7 8]_{补} = 0 1 0 0 1 1 1 0 \\ + [- 2 3]_{补} = 1 1 1 0 1 0 0 1 \\ \hline \text{进位 1 自动丢失} \longrightarrow 1 0 0 1 1 0 1 1 1 \end{array}$$

所以 $[(+78)+(-23)]_{补}=[+78]_{补}+[-23]_{补}=00110111B=[+55]_{补}$

(3) $(+78)-(+23)$

因为 $[+78]_{补}=01001110B$, $[+23]_{补}=00010111B$

$$\begin{array}{r} [[+ 2 3]_{补}]_{求补} = 1 1 1 0 1 0 0 0 + 1 = 1 1 1 0 1 0 0 1 \\ [+ 7 8]_{补} = 0 1 0 0 1 1 1 0 \\ + [[+ 2 3]_{补}]_{求补} = 1 1 1 0 1 0 0 1 \\ \hline \text{进位 1 自动丢失} \longrightarrow 1 0 0 1 1 0 1 1 1 \end{array}$$

所以 $[(+78) - (+23)]_{补}=[+78]_{补}+[[+23]_{补}]_{求补}=00110111B=[+55]_{补}$

(4) $(+78)-(-23)$

因为 $[+78]_{补}=01001110B$, $[-23]_{补}=11101001B$

$$\begin{array}{r} [[- 2 3]_{补}]_{求补} = 0 0 0 1 0 1 1 0 + 1 = 0 0 0 1 0 1 1 1 \\ [+ 7 8]_{补} = 0 1 0 0 1 1 1 0 \\ + [[- 2 3]_{补}]_{求补} = 0 0 0 1 0 1 1 1 \\ \hline 0 1 1 0 0 1 0 1 \end{array}$$

所以 $[(+78) - (-23)]_{补}=[+78]_{补}+[[- 2 3]_{补}]_{求补}=01100101B=[+101]_{补}$

从以上可以看出, 通过补码进行加减运算非常方便, 减法可转换成加法和求补运算, 得到正确的结果。

3. 十进制数在计算机内的表示

人们在日常生活中习惯使用十进制数, 但计算机内部只能处理二进制数, 为了处理方便, 在计算机中, 对于十进制数也提供了相应的二进制编码形式。

1.1.2 字符在计算机内的表示

计算机中处理的非数值信息，最重要的就是西文字符，西文字符包括字母、数字、专用字符及一些控制字符等，它们在计算机中也通过二进制编码表示，现在计算机中西文字符的编码通常采用的是美国信息交换标准代码 ASCII 码（American Standard Code for Information Interchange）。标准 ASCII 码定义了 128 个字符，用 7 位二进制数来编码，包括 26 个英文大写字母 A~Z，26 个英文小写字母 a~z、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）。计算机中一般以字节为单位，而 8 位二进制数为 1 字节，西文字符 ASCII 码通常放在低 7 位，高位一般补 0，在通信时，最高位常用作奇偶校验位。常用西文字符的 ASCII 码如表 1.2 所示。

表 1.2 常用字符的 ASCII 码(用十六进制数表示)

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
“	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

表 1.2 中，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。常用的控制符如回车键的 ASCII 码为 0DH（表中用 CR 表示），换行键的 ASCII 码为 0AH（表中用 LF 表示），空格键的 ASCII 码为 20H（表中用 SP 表示）等。

1.2 微型计算机的基本结构和工作原理

现在的计算机采用冯·诺依曼结构，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，使得能够把运算器和控制器集成在一块集成电路芯片内，我们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 Intel 公司不断开发的新型、功能强大的微处理器，推动微型计算机不断的向前发展。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 年）：4 位和低档 8 位微处理器，代表产品：4004→4040→8008。

第二代（1974—1977 年）：中高档 8 位微处理器，代表产品：Z80、I8085、MC6800，Apple-II 微机。

第三代（1978—1984 年）：16 位微处理器，代表产品：8086→8088→80286，IBM PC 系列机。1981 年 8 月 12 日，IBM 正式推出 IBM 5150，它的 CPU 是 Intel 8088，IBM 将 5150 称为 Personal Computer（个人计算机），不久，“个人计算机”的缩写“PC”成为所有个人计算机的代名词。另外，Intel 公司后来生产的微处理器都以 8086 为基础，包含 8086 指令，所以统一称为 80x86 系列。

第四代（1985—1992 年）：32 位微处理器，代表产品：80386→80486。

第五代（1993—1999 年）：超级 32 位 Pentium（奔腾）微处理器，代表产品：Pentium→Pentium II →Pentium III →Pentium 4，32 位 PC、Macintosh 机、PS/2 机。

第六代（2000 年以后）：64 位高档微处理器，代表产品：Itanium、64 位 RISC 微处理器芯片、微机服务器、工程工作站、图形工作站。

目前，Intel 生产的 CPU 一直占据市场的统治地位，也确立了 80x86 架构的行业标准。AMD 是 CPU 厂商中的后起之秀，生产的 CPU 兼容 x86 指令，与 Intel 的竞争一直没有停歇，始终都是此起彼伏。近年来尤其在桌面和笔记本电脑的市场，AMD 仍在继续提高自己的份额。微型计算机的发展趋势就是速度越来越快、容量越来越大、功能越来越强。

1.2.2 微型计算机的基本结构

微型计算机由中央处理器、存储器、输入/输出设备和系统总线等组成，典型的微型计算机基本结构如图 1.5 所示。

1. 中央处理器

微型计算机中的运算器和控制器合起来称为中央处理器（Central Processing Unit，CPU），又因为 CPU 已经能集成在一块集成电路芯片上，这就是微处理器（microprocessor），又称微处理机。CPU 是微型计算机的心脏，它的性能决定了整个微型机的各项关键指标。