

赠送  
电子课件

# 单片机原理与应用 及C51程序设计

谢维成 杨加国 主编  
董秀成 主审



清华大学出版社

第2版

# 单片机原理与应用 及C51程序设计

王 强 主编



清华大学出版社

TP368.1  
267

# 单片机原理与应用及 C51 程序设计

谢维成 杨加国 主编  
董秀成 主审

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应用系统开发的一个重要系列。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,并免费提供所有源代码和电路图的资源下载。

全书共 12 章,第 1 章介绍单片微机系统的基础知识,第 2 章介绍 MCS-51 单片机工作原理,第 3 章介绍单片机汇编程序设计,第 4 章介绍单片机 C 语言程序设计,第 5 章到第 9 章,用实例介绍 MCS-51 单片机内部资源及编程、MCS-51 单片机系统扩展、MCS-51 单片机与键盘、显示器的接口、MCS-51 单片机与 D/A、A/D 的接口和 MCS-51 单片机的其他接口,第 10 章介绍单片机应用系统设计,第 11 章介绍单片机应用系统实例,第 12 章介绍 Keil C51 集成环境的使用,附录分别提供了 MCS-51 系列单片机指令表和 C51 库函数表。

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 C 程序设计及应用”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。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

### 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;董秀成主审.—北京:清华大学出版社,2006.8  
ISBN 7-302-13349-2

I. 单… II. ①谢…②杨…③董… III. 单片微型计算机—C 语言—程序设计 IV. ①TP368.1②TP312

中国版本图书馆 CIP 数据核字(2006)第 074166 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编:100084

社 总 机:010-62770175 客 户 服 务:010-62776969

组稿编辑:彭 欣

文稿编辑:闫光龙

排 版 者:朱 康

印 装 者:三河市春园印刷有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:19.75 字数:465 千字

版 次:2006 年 8 月第 1 版 2006 年 8 月第 1 次印刷

书 号:ISBN 7-302-13349-2/TP·8318

印 数:1~4000

定 价:28.00 元

# 前 言

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。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 语言不仅学习方便，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 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。基于此，我们去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”，我们提出的实验及实践教学改革的目标是培养应用型人才。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，那么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在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，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。

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. 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源程序和单片机 C 语言源程序，并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 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。

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。

## 2. 本书内容

本书共分 12 章,具体内容如下。

第 1 章主要介绍学习单片微机系统必备的基础知识。

第 2 章介绍 MCS-51 单片机的详细工作原理。

第 3 章主要介绍包括寻址方式、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 章主要介绍单片机 C 语言程序,并列举了大量实例及详细代码。

第 5 章到第 9 章,分别用单片机汇编程序和单片机 C 语言程序以对比的方式用实例介绍 MCS-51 单片机内部资源及编程、MCS-51 单片机系统扩展、MCS-51 单片机与键盘和显示器的接口、MCS-51 单片机与 D/A 和 A/D 的接口及 MCS-51 单片机的其他接口。

第 10 章讲述单片机应用系统设计。

第 11 章介绍单片机应用系统实例。主要包括单片机应用系统的开发过程以及硬件、软件的设计技术,并以两个简单单片机应用系统设计为例讲解单片机应用系统的设计技术。

第 12 章介绍 Keil C51 集成环境的使用。

附录中提供了 MCS-51 系列单片机指令表和 C51 库函数表,以及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 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 1~12 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减少。课程学习后,可安排相应的课程设计,以便对学习内容巩固和加深理解。

另外,本书在描述中把 MCS-51 单片机常简称为“单片机”,书中采用了 Keil C51 V7.06 软件界面,读者在学习过程中也可以采用 Keil C51 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。

#### 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(用Protel 绘制,不具有电气连接功能)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5~12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。

#### 5. 致谢

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

本书第5、6章由谢维成编写,第2、3、7章由杨加国编写,第1章、第4章和附录由李茜编写,第8和第9章由赵定远编写,第10至12章由杨显富编写,最后由谢维成和杨加国统稿完成。西华大学董秀成教授在百忙中审阅了全部书稿并提出了建设性的意见。另外伍高辉、宋玉忠、郑海春、王孝平、赵华颖参与了本书部分图形的绘制工作,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。

虽然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,希望广大读者不吝指正。

编者

# 目 录

<b>第 1 章 基础知识</b> .....	1	2.4 MCS-51 系列单片机的工作方式	28
1.1 信息在计算机中的表示	1	2.4.1 复位方式	28
1.1.1 数在计算机内的表示	1	2.4.2 程序执行方式	29
1.1.2 字符在计算机内的表示	5	2.4.3 单步执行方式	29
1.2 单片机的概念及其特点	6	2.4.4 掉电和节电方式	30
1.2.1 单片机的基本概念	6	2.4.5 编程和校验方式	31
1.2.2 单片机的主要特点	6	2.5 MCS-51 系列单片机的时序	32
1.3 单片机的发展及其主要品种	7	2.5.1 机器周期和指令周期	32
1.3.1 4 位单片机	7	2.5.2 单机器周期指令的时序	32
1.3.2 8 位单片机	8	2.5.3 双机器周期指令的时序	33
1.3.3 16 位单片机	8	习题	33
1.3.4 32 位单片机	8	<b>第 3 章 单片机汇编程序设计</b>	34
1.4 单片机的应用	8	3.1 MCS-51 系列单片机汇编指令	
1.4.1 单机应用	9	格式及标识	34
1.4.2 多机应用	9	3.1.1 指令格式	34
1.4.3 单片机的等级	9	3.1.2 指令中用到的标识符	35
习题	9	3.2 MCS-51 系列单片机的寻址方式	35
<b>第 2 章 单片机基本原理</b>	11	3.2.1 常数寻址(立即寻址)	35
2.1 MCS-51 系列单片机简介	11	3.2.2 寄存器数寻址(寄存器寻址)	36
2.2 MCS-51 系列单片机的结构原理	12	3.2.3 存储器数寻址	36
2.2.1 MCS-51 系列单片机的		3.2.4 位寻址	38
基本组成	12	3.2.5 指令寻址	38
2.2.2 MCS-51 系列单片机的		3.3 MCS-51 系列单片机指令系统	39
内部结构	12	3.3.1 数据传送指令	39
2.2.3 MCS-51 系列单片机的		3.3.2 算术运算指令	42
中央处理器(CPU)	12	3.3.3 逻辑操作指令	44
2.2.4 MCS-51 系列单片机的		3.3.4 控制转移指令	45
存储器结构	15	3.3.5 位操作指令	50
2.2.5 MCS-51 系列单片机的		3.4 MCS-51 系列单片机汇编程序	
输入/输出接口	21	常用伪指令	52
2.3 MCS-51 系列单片机的外部		3.5 MCS-51 系列单片机汇编	
引脚及片外总线	25	程序设计	55
2.3.1 外部引脚	25	3.5.1 运算程序	55
2.3.2 片外总线结构	27	3.5.2 数据的拼拆和转换	58



3.5.3 多分支转移(散转)程序 .....	60	4.7.8 break 和 continue 语句 .....	90
习题 .....	62	4.7.9 return 语句 .....	91
<b>第 4 章 单片机 C 语言程序设计 .....</b>	<b>65</b>	<b>4.8 函数 .....</b>	<b>91</b>
4.1 C 语言与 MCS-51 单片机 .....	65	4.8.1 函数的定义 .....	92
4.1.1 C 语言的特点及程序结构 .....	65	4.8.2 函数的调用与声明 .....	94
4.1.2 C 语言与 MCS-51		4.8.3 函数的嵌套与递归 .....	96
单片机 .....	67	<b>4.9 C51 构造数据类型 .....</b>	<b>98</b>
4.1.3 C51 程序结构 .....	67	4.9.1 数组 .....	98
4.2 C51 的数据类型 .....	68	4.9.2 指针 .....	100
4.3 C51 的运算量 .....	70	4.9.3 结构 .....	103
4.3.1 常量 .....	70	4.9.4 联合 .....	105
4.3.2 变量 .....	72	4.9.5 枚举 .....	107
4.3.3 存储模式 .....	75	习题 .....	108
4.3.4 绝对地址的访问 .....	75	<b>第 5 章 MCS-51 单片机内部</b>	
4.4 C51 的运算符及表达式 .....	77	<b>资源及编程 .....</b>	<b>111</b>
4.4.1 赋值运算符 .....	77	5.1 并行输入/输出接口 .....	111
4.4.2 算术运算符 .....	78	5.2 定时/计数器接口 .....	111
4.4.3 关系运算符 .....	78	5.2.1 定时/计数器的主要特性 .....	111
4.4.4 逻辑运算符 .....	78	5.2.2 定时/计数器 T0、T1 的	
4.4.5 位运算符 .....	79	结构及工作原理 .....	112
4.4.6 复合赋值运算符 .....	79	5.2.3 定时/计数器的方式和	
4.4.7 逗号运算符 .....	80	控制寄存器 .....	113
4.4.8 条件运算符 .....	80	5.2.4 定时/计数器的工作方式 .....	114
4.4.9 指针与地址运算符 .....	80	5.2.5 定时/计数器的初始化	
4.5 表达式语句及复合语句 .....	81	编程及应用 .....	117
4.5.1 表达式语句 .....	81	5.3 串行接口 .....	121
4.5.2 复合语句 .....	81	5.3.1 通信的基本概念 .....	121
4.6 C51 的输入/输出 .....	82	5.3.2 MCS-51 单片机串行口	
4.6.1 格式输出函数 printf() .....	83	功能与结构 .....	123
4.6.2 格式输入函数 scanf() .....	83	5.3.3 串行口的工作方式 .....	125
4.7 C51 程序基本结构与相关语句 .....	84	5.3.4 串行口的编程及应用 .....	127
4.7.1 C51 的基本结构 .....	84	5.4 中断系统 .....	142
4.7.2 if 语句 .....	87	5.4.1 中断的基本概念 .....	142
4.7.3 switch/case 语句 .....	87	5.4.2 MCS-51 单片机的	
4.7.4 while 语句 .....	88	中断系统 .....	143
4.7.5 do...while 语句 .....	89	5.4.3 MCS-51 中断系统的	
4.7.6 for 语句 .....	89	应用 .....	147
4.7.7 循环的嵌套 .....	90	习题 .....	150

<b>第 6 章 MCS-51 单片机系统扩展</b> .....152	习题 .....	187
6.1 MCS-51 单片机的最小系统.....152		
6.1.1 8051/8751 的最小系统.....152		
6.1.2 8031 最小系统.....152		
6.2 存储器扩展 .....		153
6.2.1 存储器扩展概述.....153		
6.2.2 程序存储器扩展.....156		
6.2.3 数据存储器扩展.....158		
6.3 输入/输出口扩展 .....		159
6.3.1 简单 I/O 接口扩展 .....		159
6.3.2 可编程 I/O 扩展(8255A).....161		
习题 .....		167
<b>第 7 章 MCS-51 单片机与键盘、 显示器的接口</b> .....169		
7.1 MCS-51 单片机与键盘的接口 .....		169
7.1.1 键盘的工作原理.....169		
7.1.2 独立式键盘与单片机的 接口 .....		171
7.1.3 矩阵式键盘与单片机的 接口 .....		172
7.2 MCS-51 单片机与 LED 显示器接口 .....		177
7.2.1 LED 显示器的结构与原理 .....		177
7.2.2 LED 数码管显示器的 译码方式 .....		179
7.2.3 LED 数码管的显示方式.....180		
7.2.4 LED 显示器与单片机的 接口 .....		181
7.3 MCS-51 单片机与行程开关、 晶闸管、继电器的接口.....184		
7.3.1 行程开关、继电器与 MCS-51 单片机的接口 .....		185
7.3.2 晶闸管与 MCS-51 单片机的接口.....185		
7.3.3 继电器与 MCS-51 单片机的接口.....186		
7.3.4 蜂鸣器与 MCS-51 单片机的接口.....187		
<b>第 8 章 MCS-51 与 D/A、A/D 的接口</b> ..... 188		
8.1 MCS-51 单片机与 ADC 的接口..... 188		
8.1.1 A/D 转换器概述 .....		188
8.1.2 ADC0809 与 MCS-51 的接口..... 189		
8.2 MCS-51 单片机与 DAC 的接口..... 194		
8.2.1 D/A 转换器概述 .....		194
8.2.2 MCS-51 单片机与 8 位 DAC0832 的接口 .....		196
习题 .....		201
<b>第 9 章 MCS-51 单片机的其他接口</b> ..... 202		
9.1 LCD 与 MCS-51 接口 .....		202
9.1.1 字符型点阵式液晶显示器 .....		202
9.1.2 LCD 显示器与单片机的 接口与应用..... 207		
9.2 MCS-51 单片机与 I2C 总线 芯片接口..... 211		
9.2.1 I2C 总线简介..... 211		
9.2.2 I2C 总线 EEPROM 芯片与 单片机接口..... 213		
9.3 MCS-51 单片机与时钟日历 芯片接口..... 225		
9.3.1 并行日历时钟芯片 DS12887 与单片机接口..... 225		
9.3.2 串行日历时钟芯片与 单片机接口..... 233		
习题 .....		243
<b>第 10 章 单片机应用系统设计</b> ..... 244		
10.1 单片机应用系统的基本结构..... 244		
10.1.1 单片机应用系统的 硬件系统 .....		244
10.1.2 单片机应用系统开发的 基本过程 .....		245
10.2 单片机应用系统的硬件系统设计 ... 247		
10.2.1 硬件系统设计原则 .....		247
10.2.2 硬件设计 .....		248

10.3 单片机应用系统的软件设计 .....	249	12.3.2 如何查看和修改寄存器的内容 .....	282
10.3.1 软件设计的特点 .....	249	12.3.3 如何观察和修改变量 .....	282
10.3.2 资源分配 .....	250	12.3.4 如何观察存储器区域 .....	282
10.3.3 单片机应用系统 开发工具 .....	250	12.3.5 并行口的使用 .....	283
习题 .....	251	12.3.6 定时/计数器的使用 .....	284
<b>第 11 章 单片机应用系统设计实例</b> .....	<b>252</b>	12.3.7 串行口的使用 .....	285
11.1 单片机电子时钟的设计 .....	252	12.3.8 外中断的使用 .....	285
11.1.1 软时钟的基本原理 .....	252	习题 .....	286
11.1.2 系统硬件电路的设计 .....	252	<b>附录 A MCS-51 系列单片机指令表</b> .....	<b>287</b>
11.1.3 系统软件程序的设计 .....	253	A.1 数据传送类指令 .....	287
11.2 多路数字电压表的设计 .....	259	A.2 算术操作类指令 .....	288
11.2.1 多路数字电压表的 原理及功能 .....	259	A.3 逻辑操作类指令 .....	289
11.2.2 系统硬件电路的设计 .....	260	A.4 控制转移类指令 .....	289
11.2.3 系统软件程序的设计 .....	261	A.5 位操作类指令 .....	290
习题 .....	268	<b>附录 B C51 的库函数</b> .....	<b>292</b>
<b>第 12 章 Keil C51 集成环境的使用</b> .....	<b>269</b>	B.1 寄存器库函数 REGXXX.H .....	292
12.1 Keil C51 简介 .....	269	B.2 字符函数 CTYPE.H .....	292
12.1.1 Keil uVision2 IDE 的 安装 .....	269	B.3 一般输入/输出函数 STDIO.H .....	294
12.1.2 Keil uVision2 IDE 界面 .....	269	B.4 内部函数 INTRINS.H .....	295
12.2 Keil uVision2 IDE 的使用方法 .....	274	B.5 标准函数 STDLIB.H .....	296
12.2.1 项目文件的建立 .....	274	B.6 字符串函数 STRING.H .....	297
12.2.2 给项目添加程序文件 .....	275	B.7 数学函数 MATH.H .....	299
12.2.3 编译、连接项目, 形成 目标文件 .....	276	B.8 绝对地址访问函数 ABSACC.H .....	301
12.2.4 运行调试观察结果 .....	277	<b>附录 C 单片机技术及嵌入式 系统的网络资源</b> .....	<b>302</b>
12.2.5 多文件的处理 .....	278	C.1 单片机技术及嵌入式系统的 常见网站 .....	302
12.2.6 仿真环境的设置 .....	279	C.2 单片机技术及嵌入式系统的 官方网站 .....	302
12.3 Keil C51 的调试技巧 .....	282	<b>参考文献</b> .....	<b>303</b>
12.3.1 如何设置和删除断点 .....	282		

# 第 1 章 基础知识

## 1.1 信息在计算机中的表示

我们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,其全称为数字式电子计算机,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,先讨论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。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,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,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,其余的位用以表示数的大小。

这种连同符号位在一起作为一个数,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。

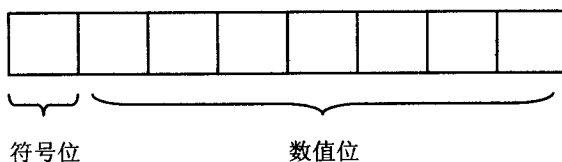


图 1.1 机器数的表示

为了运算方便,机器数在计算机中有三种表示法: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表示是相同的,负数则不是。

原码的表示如图 1.2 所示。

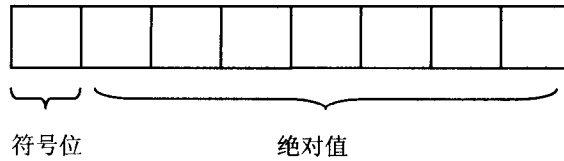


图 1.2 原码的表示

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  $-127 \sim +127$ 。

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

**【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

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

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  $X$ ， $X$  的补码也可用  $2^n - |X|$  得到，其中  $n$  为计算机字长。

**【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。

求补运算具有以下的特点：

对于一个数 X

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

那么，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。

**【例 1-4】**已知+25 的补码为 00011001B，用求补运算求-25 的补码。

因为

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

所以

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

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

补码表示时，对于-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，补码表示时运算简单。

补码的加法运算规则：

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

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

对于  $[-Y]_{\text{补}}$  只要求  $[Y]_{\text{补}}$  就可以得到。

**【例 1-5】**假设计算机字长为 8 位，完成下列补码运算。

① 25+32

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

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

② 25+(-32)

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

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

③ 25-32

$$\begin{array}{r} [25]_{补}=0011001B \quad [-32]_{补}=11100000B \\ [25]_{补}=00011001 \\ + [-32]_{补}=11100000 \\ \hline 11111001 \end{array}$$

所以  $[25-32]_{补}=[25]_{补}+[-32]_{补}=11111001B=[-7]_{补}$

④ 25-(-32)

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

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

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，得到正确的结果。

## 5. 十进制数的表示

计算机内部对信息是按二进制方式处理，但我们生活中习惯使用十进制。为了处理方便，在计算机中，对于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，分压缩 BCD 码和非压缩 BCD 码。压缩 BCD 码又称为 8421 码，它是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，编码情况见表 1.1。

表 1.1 压缩 BCD 编码表

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 124 的压缩 BCD 码为 0001 0010 0100。十进制数 4.56 的压缩 BCD 码为 0100.0101 0110。

非压缩 BCD 码是用八位二进制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 124 的非压缩 BCD 码为 0011 0001 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，还涉及到大量的字符数据。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 码(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 码见表 1.2。

计算机中一般以一个字节为单位而 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(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



## 1.2 单片机的概念及其特点

单片机作为微型计算机的一个分支,产生于 20 世纪 70 年代,经过二三十年的发展,在各行各业中已经广泛应用。单片机体积小,重量轻,抗干扰能力强,对环境要求不高,价格低廉,可靠性高,灵活性好。广泛应用于工业控制、智能仪器仪表、机电一体化产品、家用电器等领域。

### 1.2.1 单片机的基本概念

什么是单片机?单片机是微型计算机中的一种,是把微型计算机中的微处理器、存储器、I/O 接口、定时器/计数器、串行接口、中断系统等电路集成在一块集成电路芯片上形成的微型计算机。因而被称为单片微型计算机,简称为单片机。

单片机属于微型计算机的一种,它集成了微型计算机中的大部分功能部件,工作的基本原理一样,但具体结构和处理方法不同。我们知道,微型计算机由微处理器 CPU、存储器、I/O 接口三大部分通过总线有机连接而成,各种外部设备通过 I/O 接口与微型计算机连接。各个功能部件分开,功能强大。

单片机是应测控领域需要而诞生的,用以实现各种测试和控制。它的组成结构既包含通用微型计算机中的基本组成部分,又增加了具有实时测控功能的一些部件。在主芯片上集成了大部分功能部件,另外,可在外部扩展 A/D 转换器、D/A 转换器、脉冲调制器等用于测控的部件,现在一部分单片机已经把 A/D、D/A 转换器及 HSO、HIS 等外设集成在单片机中以增强处理能力。

单片机按照用途可分为通用型和专用型两大类。

(1) 通用型单片机内部资源丰富,性能全面,适应能力强。用户可以根据需要设计各种不同的应用系统。

(2) 专用型单片机是针对各种特殊场合专门设计的芯片。这种单片机针对性强,设计时根据需要设计部件。因此,它能实现系统的最简化和资源的最优化,可靠性高,成本低,在应用中有很明显的优势。

在单片机使用上注意以下几个既有相同点也有区别的概念。

(1) 单板机:将微处理器(CPU)、存储器、I/O 接口以及简单的输入/输出设备组装在一块电路板上的微型计算机,称为单板机。

(2) 单片机:将微处理器(CPU)、存储器、I/O 接口和相应的控制部件集成在一块芯片上形成的微型计算机,称为单片机。

(3) 多板机:在计算机组成中,如果组成计算机的各个功能部件是由多块电路板连接而成,这样的计算机称为多板机。

### 1.2.2 单片机的主要特点

单片机是把微处理器、存储器、I/O 接口、定时器/计数器、串行接口、中断系统等电路集成在一块集成电路芯片上形成的微型计算机。它的基本组成和基本工作原理与一般的