



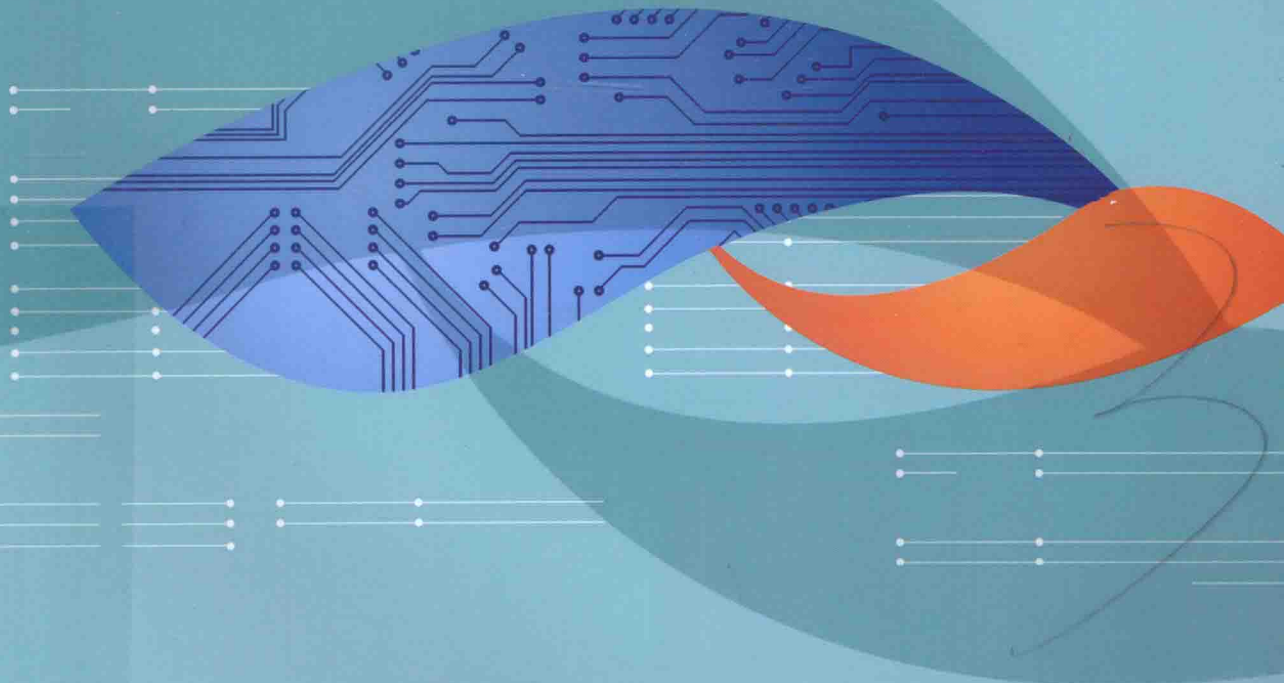
普通高等教育“十二五”规划教材



电子设计系列规划教材

单片微型计算机原理及应用 ——C语言版

◎ 姜志海 刘连鑫 赵艳雷 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

普通高等教育“十二五”规划教材
电子设计系列规划教材

单片微型计算机原理及应用 ——C 语言版

姜志海 刘连鑫 赵艳雷 编著

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书从教学的角度出发,以 51 系列单片机为硬件基础,以 C 语言为软件编程基础,系统全面地介绍 51 系列单片机的基本知识 with 基本应用,是一本重在原理与应用、兼顾理论的实用教程。本书主要内容包括:微型计算机基础、51 系列单片机硬件基础、软件编程基础——C51 语言基础、P0~P3 口应用基础、中断系统应用基础、定时器/计数器应用基础、串行口应用基础、并行总线接口扩展技术、串行总线接口扩展技术、液晶与点阵显示器应用示例、51 系列单片机应用系统设计基础等。全书提供大量的实例及详细说明与注释,硬件设计实例都经过 Proteus 仿真,每章配有本章小结、习题、实验与设计等,提供配套电子课件、程序代码、习题参考答案与实验指导。

本书可作为高等学校电子信息、自动化、计算机、电气工程、机电一体化等专业相关课程的教材,也可供相关领域科技工作者与开发人员学习参考。

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

单片微型计算机原理及应用: C 语言版 / 姜志海, 刘连鑫, 赵艳雷编著. —北京: 电子工业出版社, 2015.9
电子设计系列规划教材
ISBN 978-7-121-26190-9

I. ①单… II. ①姜… ②刘… ③赵… III. ①单片微型计算机—高等学校—教材 IV. ①TP368.1

中国版本图书馆 CIP 数据核字 (2015) 第 117519 号

策划编辑: 王羽佳

责任编辑: 王羽佳 特约编辑: 曹剑锋

印 刷: 涿州市京南印刷厂

装 订: 涿州市京南印刷厂

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: 787×1092 1/16 印张: 17.5 字数: 506 千字

版 次: 2015 年 9 月第 1 版

印 次: 2015 年 9 月第 1 次印刷

印 数: 3000 册 定价: 39.90 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 zltz@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

“微型计算机原理及应用”课程是学习和掌握微型计算机硬件基本知识和软件程序设计的入门课程,通过理论课程和实践环节的学习,掌握微型计算机的基本组成、工作原理、接口技术(硬件与软件),使学生具有应用微型计算机开发产品的初步能力。现代的微型计算机应用系统大都是采用单片机和单片机应用系统进行设计。因此,该课程是以 CPU 为核心的自动化产品的设计基础。

作为微型计算机的一个重要分支——单片机发展迅速,应用领域日益扩大,特别是在工业测控、智能仪器仪表、机电一体化产品、家电等领域得到了广泛的应用。因此,世界上许多集成电路生产厂商相继推出了各种类型的单片机,尤其是美国 Intel 公司生产的 MCS-51 系列单片机,由于其具有集成度高、处理能力强、可靠性高、系统结构简单、价格低廉、易于使用等优点,迅速占领了工业测控和自动化工程应用的主要市场,在我国也得到了广泛应用,并取得了令人瞩目的成果。MCS-51 单片机具有易于学习和掌握、性价比高优点,近年来以 MCS-51 单片机基本内核为核心的各种扩展和增强型的单片机不断推出,并且 MCS-51 单片机内核技术几乎包含了单片机理论基础和技术的全部,具有较好的系统性和完整性。尽管目前世界各大公司研制的各种高性能、不同型号的单片机不断问世,但由于国内几十年来,对于 MCS-51 单片机已积累了丰富的技术资料、完整的实验环境与开发设备,因此 51 系列单片机技术非常适合课堂教学,学懂、弄通 51 单片机的基本理论与应用技术,也就打好了学习、应用单片机的基础,这样日后学习和使用其他系列的单片机也就不难了。

在 ARM 微控制器刚推向市场时,曾有人断言,它将独占单片机市场。而几年来的市场销售情况证明,8 位字长的单片机市场主流没有发生变化,而且今后相当长一段时期内不会改变。随着单片机技术的发展,单片机功能不断增强,且由于单片机应用的规范性,目前应用于嵌入式系统的计算机内核绝大部分是单片机。所以说,单片机是构成嵌入式应用系统中最典型的主流机型。学好单片机基本理论及其技术,是开发、设计各类嵌入式应用系统的基础。

本书从教学的角度出发,以 51 系列单片机为硬件基础,以 C51 语言为软件编程基础,系统全面地介绍 51 系列单片机的基本知识 with 基本应用,是一本重在原理与应用、兼顾理论的实用教程。本书主要内容包括:微型计算机基础、51 系列单片机硬件基础、软件编程基础——C51 语言基础、P0~P3 口应用基础、中断系统应用基础、定时器/计数器应用基础、串行口应用基础、并行总线接口扩展技术、串行总线接口扩展技术、液晶与点阵显示器应用示例、51 系列单片机应用系统设计基础等。每章后附本章小结和习题以巩固所学知识。

第 2 版仍以 51 系列单片机为基础进行编写,在第 1 版基础上充实、更新内容,编排更加符合教学规律和要求,力求论述精炼、正确、由浅入深、重点突出、理论联系实际、着重应用,进一步提高全书的系统性、完整性和实用性,力争成为经典。本书具有以下鲜明的特点:

- ◎ 从零开始,轻松入门;
- ◎ 案例清晰、直观;
- ◎ 实例引导,专业经典;
- ◎ 学以致用,注重实践。

本书融入了作者多年的教学和科研经验与大量应用实例,通俗易懂、条理清晰,符合当前单片机课程的教学要求,在实例的开始进行实例分析,在实例的结束进行总结,提供详细说明和注释,并提出问题让读者思考、修改,硬件设计实例都经过 Proteus 仿真。

本书主要章节提供**实验与设计**内容。实验部分：给出实验目的、电路、基本内容、参考程序，上课教师可以根据具体情况对实验进行丰富与设计；设计题：为了锻炼学生综合分析问题与解决问题的能力，在硬件和软件上都提出设计要求，学生可以根据所学知识在硬件和软件上进行详细的设计。

本书提供教学**电子课件、程序代码、习题参考答案、实验和设计指导等配套资源**，请登录华信教育资源网 <http://www.hxedu.com.cn> 注册下载。

本书的作者都是长期使用单片机进行教学、科研和实际生产工作的教师和工程师，有着丰富的教学和实践经验。在内容编排上，按照读者学习的一般规律，结合大量实例讲解，能够使读者快速真正掌握 51 单片机的使用。本书可作为高等学校电子信息、自动化、计算机、电气工程、机电一体化等专业相关课程的教材，也可供相关领域科技工作者与开发人员学习参考。

本书由姜志海、赵艳雷、刘连鑫编写。第 1~5 章由姜志海编写，第 6、7、8、11 章由刘连鑫编写，第 9、10 章由赵艳雷编写，全书由姜志海负责整理与统稿。

本书在编写过程中得到了许多专家和同行的大力支持和热情帮助，他们对本书提出了许多建设性的建议和意见，在此一并表示衷心的感谢。

鉴于作者水平有限，加之新的单片机芯片不断涌现，其应用技术也在高速发展，书中难免有不完善和不足之处，恳请广大读者批评指正！

作 者
2015 年 9 月

目 录

第 1 章 微型计算机基础	1
1.1 微型计算机的定义与工作过程	1
1.1.1 定义	1
1.1.2 冯·诺依曼体系	1
1.1.3 工作过程	2
1.2 计算机中的数制和编码基础	3
1.2.1 计算机中的数制及转换	3
1.2.2 原码、反码、补码	3
1.2.3 定点数和浮点数	4
1.2.4 计算机中常用的编码	5
1.3 微型计算机结构	7
1.3.1 微型计算机硬件结构概述	7
1.3.2 微型计算机软件概述	8
1.3.3 CPU、存储器、I/O 口、总线	11
1.4 中断、定时器/计数器、串行通信、并行通信的初步认识	17
1.4.1 中断的初步认识	17
1.4.2 定时器/计数器的初步认识	18
1.4.3 并行通信与串行通信的初步认识	18
1.5 CPU 与外设的数据传输方式	20
1.5.1 无条件传输方式	20
1.5.2 程序查询传输方式	20
1.5.3 中断传输方式	21
1.5.4 DMA 传输方式	21
本章小结	22
习题	22
第 2 章 51 系列单片机硬件基础	23
2.1 认识单片机	23
2.1.1 单片机的特点、应用、分类、发展趋势	23
2.1.2 常用的单片机产品	27
2.1.3 MCS-51 单片机已成为国际经典	29
2.1.4 单片机与 CPU、ARM、嵌入式系统的关系	30

2.1.5 单片机应用系统开发的软硬件环境	32
2.2 51 单片机的总体结构	34
2.2.1 内部结构	34
2.2.2 外部引脚说明	35
2.2.3 CPU 的时序周期	38
2.3 51 单片机的存储器	39
2.3.1 程序存储器	39
2.3.2 数据存储器	40
2.3.3 特殊功能寄存器	42
本章小结	45
习题	45
第 3 章 51 系列单片机软件编程基础——C51 语言	46
3.1 C51 语言的数据	46
3.1.1 数据类型	46
3.1.2 常量与变量	46
3.1.3 数据存储类型	49
3.2 C51 语言对单片机主要资源的控制	51
3.2.1 特殊功能寄存器的 C51 语言定义	51
3.2.2 绝对地址的访问	52
3.2.3 位变量的 C51 语言定义	54
3.3 C51 语言的基本运算与流程控制语句	55
3.3.1 基本运算	55
3.3.2 分支判断——if、switch 语句	56
3.3.3 循环控制——while、for 语句	58
3.3.4 break、continue、return 和 goto 语句	60
3.4 C51 语言的数组、指针、函数	61
3.4.1 数组	61
3.4.2 指针	63
3.4.3 函数	65

3.5 C51 语言的预处理命令及汇编语句的嵌入	67	5.2.3 中断优先级与中断响应	104
3.5.1 文件包含、宏定义、条件编译	67	5.2.4 中断函数的结构形式	106
3.5.2 C51 中汇编语句的嵌入	69	5.3 外部中断举例	107
3.6 C51 的程序结构	70	5.3.1 外部中断源初始化	107
3.6.1 C51 的程序结构	70	5.3.2 外部中断实例	108
3.6.2 C51 编程规范及注意事项	71	5.4 实验与设计	112
3.6.3 C51 的标识符与关键字	72	本章小结	115
本章小结	74	习题	115
习题	75		
第 4 章 51 系列单片机 P0~P3 口应用基础	76	第 6 章 51 系列单片机定时器/计数器应用基础	116
4.1 认识 51 单片机的 P0~P3 口	76	6.1 可编程的硬件定时器/计数器的再认识	116
4.1.1 P1 口	76	6.1.1 功能	116
4.1.2 P3 口	77	6.1.2 工作原理	116
4.1.3 P0 口	77	6.1.3 计数器初值的计算	117
4.1.4 P2 口	79	6.2 认识 51 单片机的定时器/计数器	117
4.1.5 P0~P3 口特点总结	79	6.2.1 定时器/计数器的结构	117
4.2 输出操作	79	6.2.2 定时器/计数器的控制寄存器	118
4.2.1 基本输出操作举例——字节输出与位输出	79	6.2.3 定时器/计数器工作模式	119
4.2.2 扩展输出操作举例——流水灯与霹雳灯	80	6.3 定时器/计数器的应用举例	122
4.2.3 扩展输出操作举例——8 段 LED 静态与动态显示	82	6.3.1 定时器/计数器的初始化	122
4.3 输入操作	86	6.3.2 应用举例	124
4.3.1 闸刀型开关输入信号	87	6.4 实验与设计	131
4.3.2 单个按钮型开关输入信号	88	本章小结	134
4.3.3 多个按钮型开关输入信号——键盘	90	习题	134
4.4 实验与设计	95		
本章小结	98	第 7 章 51 系列单片机串行口应用基础	135
习题	98	7.1 串行通信的再认识	135
第 5 章 51 系列单片机中断系统应用基础	99	7.1.1 异步串行通信与同步串行通信	135
5.1 中断系统的再认识	99	7.1.2 波特率	138
5.1.1 中断的有关概念	99	7.1.3 串行通信的检错与纠错	139
5.1.2 中断处理过程	100	7.1.4 串行接口芯片 UART 和 USART	139
5.2 认识 51 单片机中断系统	102	7.2 认识 51 单片机的串行接口	140
5.2.1 51 单片机中断系统结构	102	7.2.1 串行口的结构原理	140
5.2.2 中断控制寄存器	103	7.2.2 串行口的应用控制	141
		7.3 51 单片机串行口的工作方式	143
		7.3.1 串行口工作方式 0	143
		7.3.2 串行口工作方式 1	147
		7.3.3 串行口工作方式 2	150
		7.3.4 串行口工作方式 3	153

7.4	51 单片机串行口的应用举例	157	9.1.1	认识 I ² C 总线接口	208
7.4.1	串行口编程基础	157	9.1.2	I ² C 总线典型器件 AT24C02 应用举例	210
7.4.2	串行口应用举例	159	9.2	SPI 总线接口技术	216
7.5	实验与设计	165	9.2.1	认识 SPI 总线	216
本章小结		167	9.2.2	SPI 总线典型器件 X25045 应用举例	217
习题		168	9.3	单总线 (1-wire) 接口技术	221
第 8 章	51 系列单片机并行总线接口扩展技术	169	9.3.1	认识单总线 (1-wire)	221
8.1	51 单片机并行 I/O 口扩展基础	169	9.3.2	单总线典型器件 DS18B20 应用举例	222
8.1.1	系统扩展总线结构图	169	9.4	典型串行 A/D 接口芯片 TLC2543 的 C51 编程	227
8.1.2	典型的锁存器芯片 74LS273	170	本章小结		231
8.1.3	典型的三态缓冲器 74LS244	170	习题		231
8.1.4	可编程的 I/O 接口芯片 8255A	171	第 10 章	51 系列单片机液晶与点阵显示器应用示例	232
8.2	并行总线的连接	175	10.1	51 单片机液晶显示器接口技术	232
8.2.1	数据线、控制线的连接	175	10.1.1	认识 LCD 显示器	232
8.2.2	译码信号的形成——系统扩展的寻址	176	10.1.2	字符型 LCD1602 液晶显示模块接口技术	233
8.3	并行 I/O 接口芯片扩展示例	178	10.1.3	点阵式带汉字库 12864 液晶显示模块接口技术	238
8.3.1	利用锁存器与缓冲器扩展并行的输入/输出示例	178	10.2	51 单片机点阵 LED 显示器接口技术	246
8.3.2	利用 8255A 扩展并行的输入/输出示例	180	10.2.1	认识点阵 LED 显示器	246
8.3.3	利用 8255A 作为 8 段 LED 静态显示输出示例	181	10.2.2	一个 5×7 点阵一个字符显示	247
8.3.4	利用 8255A 作为 8 段 LED 动态显示输出示例	183	10.2.3	两个 8×8 点阵字符串显示	248
8.4	模拟量接口技术	185	本章小结		249
8.4.1	A/D 与 D/A 转换器概述	185	习题		249
8.4.2	8 位并行 D/A 转换器 DAC0832 接口示例	192	第 11 章	51 系列单片机应用系统的设计	250
8.4.3	12 位并行 D/A 转换器 DAC1208 接口示例	198	11.1	单片机应用系统结构以及设计内容	250
8.4.4	8 位并行 A/D 转换器 ADC0809 接口示例	199	11.1.1	单片机应用系统的一般硬件组成	250
8.5	实验与设计	203	11.1.2	单片机应用系统的设计内容	252
本章小结		206	11.2	单片机应用系统的一般设计方法	253
习题		206	11.2.1	确定系统的功能与性能	253
第 9 章	51 系列单片机串行总线接口扩展技术	208			
9.1	I ² C 总线接口技术	208			

11.2.2	确定系统基本结构	253
11.2.3	单片机应用系统硬件与软件 设计	254
11.2.4	资源分配	256
11.3	单片机应用系统的调试	257
11.3.1	单片机应用系统调试工具	257
11.3.2	单片机应用系统的一般调试 方法	258
11.4	单片机应用系统的设计实例—— 集中供暖小型换热站控制系统 的设计	261

11.4.1	系统描述	262
11.4.2	设计方案	263
11.4.3	硬件电路设计	263
11.4.4	软件设计	267
	本章小结	267
	习题	268
附录 A	ASCII 码字符表	269
附录 B	单片机应用资料的网上查询方法	270
附录 C	Proteus 常用分离器件名称	271
	参考文献	272

第1章 微型计算机基础

进入20世纪70年代,微型计算机开始登上历史舞台,并以不可阻挡的势头迅猛发展,成为当今计算机发展的一个主流方向。当前,以微型计算机为代表的计算机已日益普及,其应用已深入社会的各个角落,极大地改变着人们的工作方式、学习方式和生活方式,成为信息时代的主要标志。本章主要概述“大学计算机基础”与“微机原理”的基本内容。

1.1 微型计算机的定义与工作过程

1.1.1 定义

以微处理器为核心,配上大容量的半导体存储器及功能强大的可编程接口芯片,连接外设(包括键盘、显示器、打印机和软驱、光驱等外部存储器)及电源所组成的计算机,称为微型计算机,简称微型机或微机,有时又称为PC(Personal Computer)或MC(Micro Computer)。微机加上系统软件,就构成了微型计算机系统(MCS,简称微机系统)。

简单地说,某系统或设备只要有CPU,就可以称为微型计算机,如平常我们说的台式机、笔记本电脑、智能洗衣机、微波炉、单片机开发板、数字式仪器仪表、机器人、自动化生产线、内部含有CPU的集成电路芯片等。

1.1.2 冯·诺依曼体系

计算机是一种能够存储程序,并能自动连续地执行程序,对各种数字化信息进行运算的现代化电子设备。

首先,计算机是能够进行各种运算的设备。运算可分为两类:算术运算和逻辑运算。算术运算的对象是数值型数据,以四则运算为基础,许多复杂的数学问题都可以通过各种算法转换成若干四则运算;逻辑运算用来解决逻辑问题,如信息检索、逻辑判断和分析等。因此,计算机的工作实际上就是对各种信息的处理。

其次,计算机如何表示这些信息呢?简单地说,是用数字代码(即二进制数)来表示各种信息,因此称为数字计算机。

最后,计算机如何对这些信息进行处理呢?它采用的是一种存储程序的工作方式,即先编写程序,再由计算机将这些程序存储起来,然后自动连续、快速地执行程序,从而实现各种运算处理。

为了存储程序与数据,需要存储器;为了进行运算,需要运算器;为了输入程序和数据及输出运算结果,需要有输入设备和输出设备;此外还需要控制器对计算机各个部件的工作进行控制和管理。

上述要领是由计算机技术的先驱冯·诺依曼提出的。他在1945年提出了数字计算机的若干思想,被称为冯·诺依曼体系,这是计算机发展史上的一个里程碑。几十年来计算机的体系结构发生了深刻的变化,但冯·诺依曼体系的核心概念仍沿用至今。冯·诺依曼体系的要点归纳如下:

- ① 采用二进制代码表示数据和指令。
- ② 采用存储程序的工作方式,即先编写程序,然后存储程序,最后自动连续地执行程序。
- ③ 计算机的硬件系统由运算器、存储器、控制器、输入设备和输出设备组成。

下面首先阐述其中两点：存储程序的工作方式、信息的数字化表示。

(1) 存储程序的工作方式

计算机的工作最终体现为执行程序，计算机采用存储程序的工作方式，体现了计算机解决问题的过程。

① 编写程序

为了使用计算机解决问题，需要先编写程序。在程序中规定了计算机需要做哪些工作，按什么步骤去做。程序还包括需要处理的原始数据，此外还规定了计算机何时从输入设备去获取数据。一件事情一般要分成几步来完成，每步执行的操作命令称为一条指令。计算机最终执行的程序是一系列指令序列，即若干指令的有序集合。换言之，我们事先编写的程序最终变成指令序列和原始数据。

② 存储程序

编写完成的程序经输入设备送入计算机，存放在存储器中。编写程序时是用字符书写的，通过键盘将字符变成二进制编码，然后再送入计算机。

③ 自动、连续地执行程序

由于程序已存储在存储器中，启动计算机后，计算机就可以按照一定的顺序从存储器中逐条读取数据，按照指令的要求完成相应的操作，直到程序被执行完毕。原则上，程序在执行过程中不需要人工干预。当然，有些工作本身需要以人机对话的形式进行，例如通过计算机进行查询时，计算机通过屏幕向操作人员询问，操作人员通过键盘或鼠标进行选择。这种情况要求计算机能分段执行程序，之间允许用户进行人工干预。所以计算机在自动、连续地执行程序的过程中，往往允许使用者以外部请求方式进行干预。

(2) 信息的数字化表示

上面讲到，现在广泛使用的计算机，其全称是电子式数字计算机。

“电子”指计算机的主要部件是由电子电路组成的，计算机内传送与处理的信息是电子信号。例如，计算机中的算术运算单元（ALU）主要由加法器构成，而加法器由各种门电路（与门、非门等）组成。

“数字”则表示计算机中的信息（控制信息和数据信息）均采用数字化表示方法。例如，二进制11001表示-9，01000001表示字符A等。

1.1.3 工作过程

计算机的工作过程实际上是执行程序的过程，而程序是由一系列指令组成的，因此执行程序的过程就是按顺序执行指令的过程。

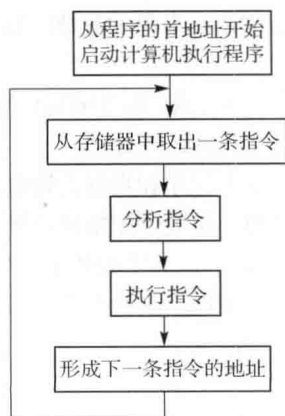


图 1-1 程序的执行过程

通常，计算机要运行某个程序时，该程序预先要调入内存的一系列单元中，在程序执行过程中完全由计算机自动执行而不需要人工干预，包括以下步骤，具体过程如图 1-1 所示。

①取出指令：从存储器某个地址中取出要执行的指令送到 CPU 内部的指令寄存器暂时保存。

②分析指令：把指令寄存器中的指令送到指令译码器，分析出该指令对应的操作。

③执行指令：根据指令译码结果，向各个部件发出相应的控制信号，完成指令规定的各种操作。

④形成下一条指令地址：为执行下一条指令做好准备，即形成下一条指令的地址。

1.2 计算机中的数制和编码基础

计算机的工作过程就是对数据进行处理。计算机是一个典型的数字化设备,它只能识别 0 和 1,所有的计算机都是以二进制数的形式进行算术运算和逻辑操作的。

1.2.1 计算机中的数制及转换

1. 计算机中的数制

计算机最早是作为一种计算工具出现的,所以最基本的功能是对数进行加工和处理。在使用微型计算机时常用的计数制有二进制数、十六进制数、十进制数 3 种。

(1) 十进制数 (Decimal)

以 10 为基数的计数制称为十进制计数制。十进制数有如下两个特点:

① 有 0~9 共 10 个不同的数码。

② 在加法中采用逢 10 进 1 的原则。

人们在实际生活中常用的是十进制数。

(2) 二进制数 (Binary)

以 2 为基数的计数制称为二进制计数制。二进制数有如下两个特点:

① 有 0、1 共 2 个不同的数码。

② 在加法中采用逢 2 进 1 的原则。

计算机中使用的是二进制数。

(3) 十六进制数 (Hexadecimal)

十六进制是人们学习和研究计算机中二进制数的一种工具,它是随着计算机的发展而广泛应用的。十六进制数有如下三个特点:

① 有 0~9、A、B、C、D、E、F 共 16 个不同的数码。

② 在加法中采用逢 16 进 1 的原则。

③ 在书写一个十六进制数时,如果该数据的高位为 A~F,则在高位的前面加“0”,如 0AH、0EFH、0FEABH、0CB98H 等。

十进制数、二进制数和十六进制数之间的关系如表 1-1 所示。

表 1-1 十进制数、二进制数及十六进制数对照表

十进制	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
二进制	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
十六进制	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

为了区别十进制数、二进制数及十六进制数 3 种数制,可以在数的后面加一个字母。规定 B(binary)表示二进制数, D(decimal)表示十进制数, H(hexadecimal)表示十六进制数,其中十进制数后面的字母 D 可以省略。

1.2.2 原码、反码、补码

数据在计算机内采用符号数字化处理后,机器可表示并识别带符号的数据。为了改进运算方法、简化控制电路,人们研究出多种有符号数的编码形式,最常用的有三种方法,即原码、反码、补码表示法。

8 位二进制数用来表示的无符号数、原码、反码和补码，如表 1-2 所示。

从表 1-2 可以看出，8 位二进制数无符号数表示范围是 0~255，有符号数原码表示范围是-127~+127，反码表示范围是-127~+127，补码表示范围是-128~+127。

表 1-2 原码、反码和补码表

二进制数	无符号数	有符号数		
		原 码	反 码	补 码
0000 0000	0	+0	+0	+0
0000 0001	1	+1	+1	+1
0000 0010	2	+2	+2	+2
...				
0111 1110	126	+126	+126	+126
0111 1111	127	+127	+127	+127
1000 0000	128	-0	-127	-128
1000 0001	129	-1	-126	-127
...				
1111 1101	253	-125	-2	-3
1111 1110	254	-126	-1	-2
1111 1111	255	-127	-0	-1

1.2.3 定点数和浮点数

计算机中运算的数有整数，也有小数。通常有两种规定：一种是规定小数点的位置固定不变，这时的机器数称为定点数；另一种是小数点的位置可以浮动，这时的机器数称为浮点数。微型计算机中常使用定点数。

1. 定点数

所谓定点法，是指小数点在数中的位置是固定不变的，以定点法表示的实数称为定点数。根据小数点位置的固定方法不同，又可分为定点整数和定点小数表示法。前面介绍的整数均为定点整数，可以认为小数点固定在数的最低位之后。

如果小数点隐含固定在整个数值的最右端，符号位右边所有的位数表示的是一个整数，即为定点整数。例如，对于 16 位机，如果符号位占一位，数值部分占 15 位，于是机器数 0111111111111111 的等效十进制数为+32767，其符号位、数值部分、小数点的位置示意如图 1-2 所示。

如果小数点隐含固定在数值的某一位置上，即为定点小数。如果小数点固定在符号位之后，即为纯小数。假设机器字长为 16 位，符号位占一位，数值部分占 15 位，于是机器数 1 000000000000001 的等效十进制数为-2⁻¹⁵，其符号位、数值部分、小数点的位置示意图如图 1-3 所示。

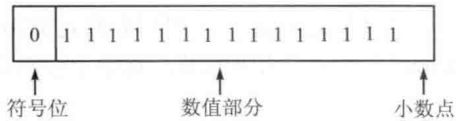


图 1-2 定点整数的符号位、数值部分和小数点位置示意图

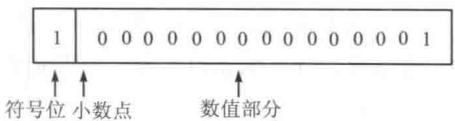


图 1-3 纯小数的符号位、数值部分和小数点位置示意图

2. 浮点数

所谓浮点数，是指计算机中数的小数点位置不是固定的，或者说是“浮动”的。在计算机中，浮

点数法一般用来表示实数,可以采用“阶码表示法”来表示浮点数,它由整数部分和小数部分组成,一个实数可以表示成一个纯小数和一个乘幂之积。

采用浮点数最大的特点是,比定点数表示的范围大。

例如,对于十进制数 $56.725=10^2 \times 0.56725$; 对于二进制数 $110.11=2^2 \times 1.1011$ 。

对于任何一个二进制数 N , 都可以表示为

$$N=(2^{\pm E}) \times (\pm S)$$

浮点数在计算机中的编码基本格式如图 1-4 所示。

阶符	阶码 $E(m \text{ 位})$	尾符	尾数编码 $S(n \text{ 位})$
----	---------------------	----	-----------------------

图 1-4 浮点数在计算机中的编码基本格式

其中, E 称为阶码, 阶码为 0 表示 E 为正, 为 1 表示 E 为负。由此可见, 小数点的实际位置随着阶码 E 的大小和符号而浮动决定; $\pm S$ 为全部有效数据, 称为尾数部分。

例如, $1001.011=2^{0100} \times (0.1001011)$ 。此处, 0100 部分称为阶码且为正, (0.1001011) 部分称为尾数。

浮点数的格式多种多样。例如, 某计算机用 4 个字节表示浮点数, 阶码部分为 8 位补码定点整数, 尾数部分为 24 位补码定点小数, 如图 1-5 所示。



图 1-5 4 个字节表示的浮点数

【例 1-1】 描述用 4 个字节存放十进制浮点数“136.5”的浮点格式。

由于 $(136.5)_{10}=(10001000.1)_2$, 将二进制数“10001000.1”进行规格化, 即

$$10001000.1=0.100010001 \times 2^8$$

阶码 2^8 表示阶符为“+”, 阶码为“8”的二进制数为“0001000”; 尾数中的数符为“+”。小数值为“100010001”。

十进制小数“136.5”在计算机中的表示如图 1-6 所示。

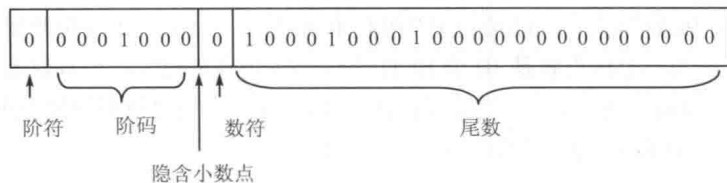


图 1-6 规格化后的浮点数

在实际应用时, 由于阶码指数可使用不同的编码 (原码、补码等), 尾数的格式和小数点的位置也可以有不同的规定, 所以浮点数的表示方法不唯一, 不同的计算机可以有不同的规定。

1.2.4 计算机中常用的编码

计算机除了用于数值计算外, 还要进行大量的文字信息处理, 也就是要对表达各种文字信息的符号进行加工。例如计算机和外设如键盘、(字符) 显示器、打印机之间的通信都采用字符方式输入/输出。目前计算机中最常用的两种编码是美国信息交换标准代码 (ASCII 码) 和二—十进制编码 (BCD 码)。

1. 美国信息交换标准代码 (ASCII 码)

ASCII (American Standard Code for Information-Interchange) 码是美国信息交换标准代码的简称, 主要给西文字符进行编码。它采用 7 位二进制数表示一个字符, 包括 32 个标点符号, 10 个阿拉伯数字, 52 个英文大小写字母, 34 个控制符号, 共 128 个。编码与字符之间的对应关系如附录 A 所示。

在计算机系统中, 存储单元的长度通常为 8 位二进制数 (即一个字节), 为了存取方便, 规定一个存储单元存放一个 ASCII 码, 其中低 7 位表示字母本身的编码, 第 8 位 (即 bit7) 用作奇偶校验位或规定为零 (通常如此)。因此, 也可以认为 ASCII 码的长度为 8 位。

奇偶校验的主要目的是用于在数据传输过程中, 检测接收方的数据是否正确。收发双方预约为何种校验, 接收方收到数据后检验 1 的个数, 判断是否与预约的校验相符, 倘若不符, 则说明传输出错, 可请求重新发送。奇校验时, bit7 的取值应使得 8 位 ASCII 码中 1 的个数为奇数; 偶校验时, bit7 的取值应使得 ASCII 码中 1 的个数是偶数。例如:

“8”的奇校验 ASCII 码为 00111000B, 偶校验 ASCII 码为 10111000B。

“B”的奇校验 ASCII 码为 11000010B, 偶校验 ASCII 码为 01000010B。

2. BCD 码 (二进制编码的十进制数)

十进制毕竟是人们最习惯的计数方式, 在向计算机输入数据时, 常用十进制数输入, 但计算机只识别二进制数, 因此每 1 位十进制必须用二进制数表示。1 位十进制数包含 0~9 十个数码, 必须用 4 位二进制数表示, 这样就需要确定 0~9 与 4 位二进制数 0000B~1111B 之间的对应关系, 其中较常用的 8421BCD 码规定了十进制数 0~9 与 4 位二进制数编码之间的对应关系, 见表 1-3。

表 1-3 十进制数与 4 位二进制数编码之间的对应关系

十进制数	8421BCD 码	十进制数	8421BCD 码
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

注: 在 BCD 码中, 不使用 1010B (0AH) ~ 1111B (0FH)。

例如, 4567.89 的 BCD 码为 0100 0101 0110 0111.1000 1001 (每 1 位十进位数用相应的 4 位二进制数表示即可)。BCD 码的一个优点就是 10 个 BCD 码组合格式容易记忆。一旦熟悉了 4 位二进制数的表示, 对 BCD 码就可以像十进制数一样迅速自如地读出。同样, 也可以很快地得出以 BCD 码表示的十进制数。例如, 将 1 个 BCD 数转换成相应的十进制数:

$$(0111\ 0110\ 1001.1001\ 0011\ 0101)_{\text{BCD}} = 769.935$$

BCD 码采用 4 位编码, 4 位一组表示 1 位十进制数, 分别表示十进制数的个位、十位、百位等, 低 4 位对高 4 位的进位为“逢十进一”。它不是二进制数, 而是按 4 位二进制数的展开值和按 4 位一组解释为十进制数。BCD 码和二进制数之间的转换不能直接实现, 二进制数转换为 BCD 码时, 必须先将其转换为十进制数。例如, 要将二进制数 1101.1B 转换为 BCD 码, 首先需要将二进制数按权展开, 转换为十进制数, 再转换为 BCD 码:

$$1101.1\text{B} = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} = 13.5$$

$$13.5 = (0001\ 0011.0101)_{\text{BCD}}$$

同样, 要将 BCD 码转换为二进制数, 首先要将 BCD 码转换为十进制数, 然后再转换为二进制数。

BCD 编码可以简化人机关系,但它比纯二进制编码的效率低。对同一个给定的十进制数,用 BCD 编码表示的位数比用纯二进制表示的位数要多。而每位数都需要某些数字电路与之对应,这就使得与 BCD 码连接的附加电路成本提高,设备的复杂性增加,功耗较大。用 BCD 码进行运算所花的时间比用纯二进制码进行运算所花的时间要多,而且复杂。

计算机中存储 BCD 码的形式有两种：压缩 BCD 码和非压缩 BCD 码。

(1) 压缩 BCD 码

压缩BCD码用4位二进制数表示1位十进制数,一个字节可以表示2位十进制数。例如10010111B表示十进制数97。

(2) 非压缩 BCD 码

非压缩BCD码用8位二进制数表示1位十进制数,高4位总为0000,低4位的0000~1001表示0~9。例如00001001B表示十进制数9。

尽管 BCD 码比较直观, 但 BCD 码与二进制数之间的转换并不方便, 需要转换成十进制数后, 才能转换为二进制数, 反之亦然。

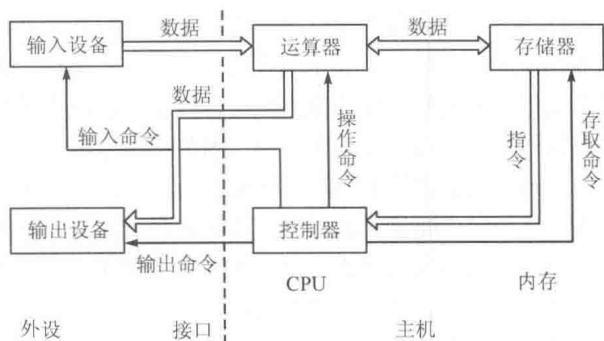
前面介绍了在使用计算机时二进制数、十进制数、十六进制数、ASCII 码、BCD 码以及带符号数的表示等问题,这里要注意微型计算机能处理的数据只有二进制数,计算机并不认识什么正数、负数、BCD 码、ASCII 码等,计算机中数的表现形式只有二进制数,其他的数制和性质需要人们来进行分析与说明。如在某存储器中存放一个二进制数 11111111B (0FFH),这个数多大?这要看人们如何看了,如果是一个无符号数,就是 255;如果是一个有符号数,就是-1;如果是个 BCD 码,就是一个无效的数;如果是个 ASCII 码,就代表“DEL”键的 ASCII 码值。

1.3 微型计算机结构

微型计算机是由硬件（Hardware）和软件（Software）两大部分组成的。硬件是由电子部件和机电装置所组成的计算机实体，其基本功能是接收计算机程序，并在程序控制下完成信息输入、处理和结果输出等任务。软件是指为计算机运行服务的全部技术资料和各种程序，以保证计算机硬件的功能得以充分发挥。

1.3.1 微型计算机硬件结构概述

微型计算机在硬件上由运算器、控制器、存储器、输入设备及输出设备五大部分组成,如图 1-7 所示。



运算器是计算机处理信息的主要部分；控制器控制计算机各部件自动地、协调一致地工作；存储器是存放数据与程序的部件；输入设备用来输入数据与程序，常用的输入设备有键盘、光电输入机等；输出设备将计算机的处理结果用数字、图形等形式表示出来，常用的输出设备有显示终端、数码管、打印机、绘图仪等。

通常把运算器、控制器、存储器这三部分称为计算机的主机，而输入、输出设备则称为计算机的外部设备（简称外设）。由于运算器、控制器是计算机处理信息的关键部件，所以常将它们合称为中央

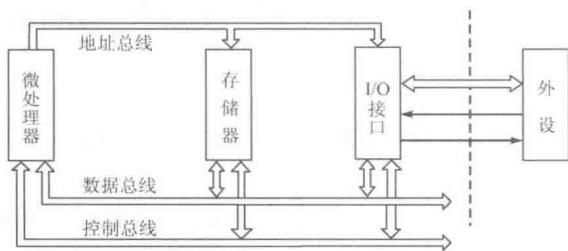


图 1-8 微型计算机结构

处理单元 CPU（Central Processing Unit）。将微处理器、存储器、I/O 接口电路集成在一块芯片上，称为单片微型计算机，简称单片机。

关于 CPU、存储器、I/O 口、总线的有关问题将在下节讨论。

1.3.2 微型计算机软件概述

上面所述的微型计算机设备称为硬件。计算机能够脱离人的直接控制而自动地操作与运算，还必须要软件。软件是指使用和管理计算机的各种程序（Program），而程序是由一条条指令（Instruction）组成的。

控制计算机完成各种操作的命令称为指令。指令分成操作码和操作数两部分。操作码表示该指令执行何种操作，操作数表示参加运算的数据或数据所在的地址。

为了计算一个数学式，或者控制一个生产过程，需要事先制定计算机的计算步骤或操作步骤。计算步骤是由一条条指令来实现的。这种一系列指令的有序集合称为程序。编制程序的过程称为程序设计。

为了使机器能自动进行计算，要预先用输入设备将程序输入计算机存放。计算机启动后，在控制器的控制下，CPU 按照顺序依次取出程序的一条条指令，加以译码和执行。计算机的工作是由硬件、软件紧密结合、共同完成的，这与一般的数字电路系统不同。

1. 源程序与机器码

在程序设计时，要编写源程序。编写源程序可以采用符号语言或者是目标语言。目标语言是机器码语言，现在通常采用符号语言。符号语言基本可以分为汇编语言和高级语言。

用助记符（通常是指令功能的英文缩写）表示操作码，用字符（字母、数字、符号）表示操作数的指令称为汇编指令。用汇编指令编制的程序称为汇编语言程序。这种程序占用存储器单元较少，执行速度较快，能够准确掌握执行时间，可实现精细控制，因此特别适用于实时控制。然而汇编语言是面向机器的语言，各种计算机的汇编语言是不同的，必须对所用机器的结构、原理和指令系统比较清楚才能编写出它的各种汇编语言程序，而且不能通用于其他机器，这是汇编语言的不足之处。

高级语言是面向过程的语言，常用的高级语言有 BASIC、FORTRAN、PASCAL、C 等。用高级语言编写程序时主要着眼于算法，而不必了解计算机的硬件结构和指令系统，因此易学易用。高级语言是独立于机器的，一般来说，同一个程序可在任何种类的机器中使用。高级语言适用于科学计算、数据处理等。C 语言是一种编译型程序设计语言，它兼顾了多种高级语言的特点，并具备了汇编语言