

高等学校电子信息类专业
“十二五”规划教材

ELECTRONIC
INFORMATION SPECIALTY

微控制器原理 及应用

主 编 张晓莉
副主编 何 蓉 朱贵宪 吴文峰

 西安电子科技大学出版社
<http://www.xduph.com>



014033445

TP332.3
372

高等学校电子信息类专业“十二五”规划教材

内容简介

微控制器原理及应用

主 编 张晓莉
副主编 何 蓉 朱贵宪 吴文峰
参 编 朱代先 邵小强



西安电子科技大学出版社



北航

C1721997

TP332.3
372

内 容 简 介

本书是按照教育部关于电子类、电气类专业应用型人才培养计划的基本要求,结合当前微控制器的发展状况而编写的,主要介绍以 MCS-51 单片机为主的微控制器的基本原理及应用技术,内容涵盖了微型计算机原理和微控制器的应用两部分,包括计算机的数制及其转换,微型计算机结构, MCS-51 系列单片机结构,指令系统及汇编语言程序设计,中断、定时/计数器与串行口, C51 语言程序设计基础,单片机系统的扩展,基于 MCS-51 的典型串行总线设计等。书中最后一章以典型工业检测及控制产品的设计为例,介绍了微控制器系统的开发过程及步骤,并提供了详细的源代码。本书内容详实、由浅入深、图文并茂,理论教学与实践讲解相结合,重点放在计算机基础知识的学习和嵌入式基本应用技能的培养上。

本书可作为高等学校和各类技术院校自动化专业、通信专业、电子技术应用专业及计算机专业在校学生的教材,也可作为自学和从事微控制器研发工作的工程技术人员的参考用书。

图书在版编目(CIP)数据

微控制器原理及应用/张晓莉主编. —西安:西安电子科技大学出版社, 2014.2

高等学校电子信息类专业“十二五”规划教材

ISBN 978-7-5606-3319-0

I. ① 微… II. ① 张… III. ① 微控制器—高等学校—教材 IV. ① TP332.3

中国版本图书馆 CIP 数据核字(2014)第 018631 号

策 划 云立实

责任编辑 云立实 高丽萍

出版发行 西安电子科技大学出版社(西安市太白南路2号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 陕西天意印务有限责任公司

版 次 2014年2月第1版 2014年2月第1次印刷

开 本 787毫米×1092毫米 1/16 印张 25

字 数 595千字

印 数 1~3000册

定 价 43.00元

ISBN 978-7-5606-3319-0/TP

XDUP 3611001-1

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜,谨防盗版。

前 言

微控制器以成本低、功能强、简单易学、使用方便等独特的优势,在智能仪表、工业测控、数据采集、计算机通信等各个领域得到了极为广泛的应用。目前,以单片机为核心的微控制器开发技术已经成为电子信息、自动化、通信、电气、机电一体化、计算机应用等专业学生、相关专业技术人员必须掌握的技术之一。

在对单片机系列课程改革的前提下,本书针对微机原理及单片机课程理论多、应用性不强的特点,将微机原理、单片机及嵌入式系统课程整合,以培养能力、突出实用为基本出发点,重点讲解了 MCS-51 单片机基本概念、基本知识点。本书还结合不同的实例,以实用技术为主线,在简单介绍微型计算机原理的基础上,详细介绍了单片机的原理和应用,并与后续嵌入式系统设计紧密衔接。

本书共分为 9 章,从四个方面介绍了微控制器的原理及应用:首先介绍了计算机的数制及其转换,微型计算机的结构;其次以 51 系列单片机为主介绍了微控制器的结构及操作指令;再次介绍了单片机的特殊功能部件及 C51 语言;最后以工业控制单元、家用电器和无线网络控制器为例介绍了单片机应用的开发过程,并给出了典型实例的设计步骤及主要程序的源代码。

本书由张晓莉担任主编,何蓉、朱贵宪和吴文峰担任副主编。本书得到了陕西省教改项目——基于 CDIO 的电子信息技术类应用型人才培养模式的研究与实践,和陕西省精品课程微机应用系统设计项目的支持。本书在编写过程中,得到了西安科技大学吴延海教授等老师的指导,吴教授还审阅了书稿,在此表示衷心的感谢!

由于计算机技术发展迅速,多媒体应用软件日益更新,加上作者水平有限,疏漏之处在所难免,恳请广大读者批评指正。

编 者

2013 年 8 月

目 录

第1章 计算机的数制及其转换	1	3.2 中央处理器 CPU	75
1.1 计算机的数制及其转换	1	3.2.1 运算器	75
1.1.1 数与数制	2	3.2.2 控制器及振荡器	76
1.1.2 不同数制间的转换	5	3.2.3 位(布尔)处理器	77
1.2 计算机中数与字符的编码	15	3.3 存储器	77
1.2.1 数值数据的编码及运算	15	3.3.1 存储器的分类及存储空间配置	77
1.2.2 非数值数据的二进制编码	32	3.3.2 内部数据存储器	78
1.3 微型计算机的性能分析及分类	34	3.3.3 外部数据存储器	83
1.3.1 微型计算机的性能分析	34	3.3.4 程序存储器	84
1.3.2 微型计算机的分类	37	3.3.5 Flash 闪速存储器的编程	85
习题	40	3.4 并行输入/输出接口	86
第2章 微型计算机结构	41	3.4.1 I/O 接口电路概述	86
2.1 微型计算机概念	41	3.4.2 P0 口	86
2.1.1 微型计算机的历史	41	3.4.3 P1 口	87
2.1.2 计算机的基本模型	42	3.4.4 P2 口	88
2.2 微型计算机的组成	45	3.4.5 P3 口	88
2.2.1 微型计算机的硬件	46	3.4.6 I/O 端口负载能力	89
2.2.2 微型计算机的软件	48	3.5 I/O 接口电路的作用与 I/O 接口的 编址方式	89
2.2.3 微型计算机系统的主要技术指标	48	3.5.1 I/O 接口电路的作用	90
2.3 微处理器的结构及原理	49	3.5.2 I/O 接口的编址方式	91
2.3.1 中央处理器的发展过程	49	3.6 CPU 的时序与复位	91
2.3.2 存储系统	56	3.6.1 时序的基本概念	92
2.3.3 当前微处理器所使用的先进技术	61	A 3.6.2 CPU 的时序	93
2.4 单片机概述	63	3.6.3 复位电路与复位状态	93
2.4.1 单片机基本概念	63	3.6.4 掉电与节电方式	94
2.4.2 单片机的产生	64	习题	95
2.4.3 单片机的应用	65	第4章 指令系统及汇编语言程序 设计	96
2.4.4 单片机的发展趋势	66	4.1 程序设计概述	96
2.5 单片机主要系列	67	4.1.1 指令和程序设计语言	96
2.5.1 主要生产制造商及其特点	67	4.1.2 汇编概念	98
2.5.2 单片机的四个主要系列	68	4.2 指令格式和寻址方式	98
习题	70	4.2.1 指令格式	98
第3章 MCS-51 系列单片机结构	71	4.2.2 指令中常用符号	100
3.1 MCS-51 单片机的结构和引脚	71	4.2.3 寻址方式	100
3.1.1 单片机的内部结构及功能部件	71		
3.1.2 单片机外部引脚说明	73		

4.3 MCS-51 单片机指令系统	104	6.1.1 C51 程序创建过程	193
4.3.1 数据传送类指令	104	6.1.2 存储空间定义	195
4.3.2 算术运算类指令	113	6.1.3 C51 数据类型	196
4.3.3 逻辑运算及移位类指令	118	6.1.4 C51 的常量	197
4.3.4 控制转移类指令	121	6.1.5 C51 常用的运算符	197
4.3.5 位操作指令	128	6.1.6 C51 的表达式	199
4.4 汇编语言程序设计	132	6.1.7 C51 的基本语句	200
4.4.1 伪指令	132	6.2 C51 的函数和数组	200
4.4.2 汇编语言源程序格式	135	6.2.1 函数的定义	200
4.4.3 汇编语言程序的设计步骤与基本 结构	136	6.2.2 数组的定义	201
4.4.4 顺序结构程序设计	139	6.3 C51 的编程规范	201
4.4.5 分支(选择)结构程序设计	139	6.3.1 注释	201
4.4.6 循环结构程序设计	142	6.3.2 命名	201
4.4.7 常用子程序设计	145	6.3.3 格式	201
4.5 汇编语言应用程序的开发与调试	150	6.4 C51 的基本运用	201
4.5.1 仿真开发系统简介	150	第7章 单片机系统的扩展	209
4.5.2 程序的开发调试过程	151	7.1 概述	209
习题	152	7.2 系统总线扩展	210
第5章 中断、定时/计数器与串行口 ..	155	7.2.1 系统总线扩展简介	210
5.1 中断	155	7.2.2 常用扩展器件介绍	212
5.1.1 中断系统概述	155	7.2.3 编址技术	213
5.1.2 MCS-51 单片机的中断系统	157	7.3 存储器的扩展	214
5.1.3 中断处理过程	162	7.3.1 存储器扩展概述	214
5.1.4 中断请求的撤除	164	7.3.2 存储器扩展应考虑的问题	217
5.1.5 中断程序应用举例	165	7.3.3 程序存储器的扩展	217
5.2 定时/计数器	167	7.3.4 数据存储器的扩展	220
5.2.1 定时/计数器的结构及工作原理 ..	168	7.4 I/O 接口的扩展及应用	222
5.2.2 定时/计数器的控制与实现	170	7.4.1 I/O 接口电路的作用	222
5.2.3 定时/计数器的工作方式	171	7.4.2 接口与端口	223
5.2.4 定时/计数器的应用举例	173	7.4.3 I/O 的传送方式	224
5.3 串行接口	176	7.4.4 用 TTL 芯片扩展 I/O 口	225
5.3.1 串行通信的基本概念	177	7.4.5 用可编程芯片扩展 I/O 口	226
5.3.2 MCS-51 串行口的结构与工作 原理	179	7.5 LED 数码显示器接口	233
5.3.3 串行口控制寄存器	181	7.5.1 LED 数码显示器的结构与连接 方式	233
5.3.4 串行口的工作方式	182	7.5.2 LED 数码显示器的接口电路与 显示方法	235
5.3.5 MCS-51 串口的波特率	185	7.5.3 LED 数码显示器应用举例	238
5.3.6 MCS-51 多机通信技术	187	7.6 键盘接口	243
5.3.7 串行口的应用举例	188	7.6.1 键盘接口的工作原理	243
习题	191	7.6.2 键盘接口电路	244
第6章 C51 语言程序设计基础	193	7.6.3 键盘扫描程序	248
6.1 C51 语言基础	193	7.6.4 键盘接口设计实例	248

7.7 A/D 转换器.....	258	第 9 章 单片机应用系统设计与应用	
7.7.1 A/D 转换器原理及性能指标.....	258	实例	309
7.7.2 典型 A/D 转换器芯片 ADC0809 简介	261	9.1 应用系统设计原则	309
7.7.3 MCS-51 单片机与 ADC0809 接口	263	9.2 应用系统设计流程	310
7.7.4 A/D 转换应用举例.....	266	9.2.1 总体设计	310
7.7.5 串行 A/D 转换接口芯片 TLC2543	266	9.2.2 硬件电路设计	311
7.8 D/A 转换器.....	270	9.2.3 印刷电路板设计	312
7.8.1 D/A 转换器的原理及性能指标.....	270	9.2.4 软件程序设计	312
7.8.2 典型 D/A 转换器芯片 DAC0832 介绍	272	9.2.5 调试、运行与维护	313
7.8.3 MCS-51 单片机与 DAC0832 接口	273	9.3 应用系统的可靠性及抗干扰设计	313
7.8.4 串行 D/A 转换接口芯片 TLC5615	277	9.3.1 干扰来源	314
习题	279	9.3.2 电源系统抗干扰设计	314
第 8 章 基于 MCS-51 的典型串行总线 设计	280	9.3.3 地线干扰及抑制	315
8.1 概述	280	9.3.4 其它提高系统可靠性的方法	316
8.2 RS-485 总线	280	9.4 应用系统实例 1——温度时间测量与 显示系统	316
8.2.1 RS-485 电气特性	281	9.4.1 系统的功能分析	317
8.2.2 RS-485 总线节点数和通信 方式	282	9.4.2 系统的设计方案	317
8.2.3 RS-485 总线收发器与单片机接口 电路的设计	283	9.4.3 系统硬件电路的设计	317
8.2.4 基于单片机节点的 RS-485 总线的 工作原理	285	9.4.4 控制过程的软件程序实现	328
8.3 SPI 总线	293	9.5 应用系统实例 2——太阳能电池板追踪 系统	337
8.3.1 SPI 总线的工作原理	293	9.5.1 系统的功能分析	337
8.3.2 SPI 总线的通信时序	294	9.5.2 系统的设计方案	338
8.3.3 硬件电路设计	295	9.5.3 系统硬件电路的设计	338
8.3.4 软件程序设计	298	9.5.4 系统控制过程的软件设计	342
8.4 I ² C 总线	299	9.6 应用系统实例 3——基于 GSM 网络的 远程遥测系统设计	348
8.4.1 I ² C 总线的工作原理	299	9.6.1 系统功能需求分析	348
8.4.2 I ² C 总线的通信时序	302	9.6.2 系统方案设计	349
8.4.3 硬件电路设计	303	9.6.3 系统硬件电路设计	349
8.4.4 软件程序设计	303	9.6.4 基于 GSM 网络的远程遥测系统的 软件设计	363
习题	308	附录	386
		附录 I ASCII 码表	386
		附录 II ASCII 码符号说明	387
		附录 III MCS-89C51 系列单片机指令表	387
		参考文献	391

第1章 计算机的数制及其转换

教学提示：任何数据在计算机中都用二进制表示，而数据又有数值数据和非数值数据两种。数值数据常用的编码有原码、反码和补码。由于补码编码有许多优点，因此大多数微机数字与字符采用补码进行编码。在计算机内部的十进制数的编码通常是BCD码。对于非数值数据，英文字母及符号通常用ASCII编码，而汉字则需要两个字节来进行编码。数制及其转换、数与字符的编码是计算机的基础。

教学要求：本章主要介绍计算机的数制及其转换，数与字符的编码，微型计算机的性能等相关知识。通过本章的学习使学生了解微型计算机的特点、分类、主要技术指标；掌握计算机中的数制及其转换，计算机数据和字符的编码。

电子计算机(Computer)是20世纪人类最重要的科技发明与成果之一。计算机是一种能够自动、高速、精确地进行信息处理的现代化电子设备，具有算术运算和逻辑判断能力，并能通过预先编好的程序来自动完成数据的处理。自1946年世界上第一台计算机问世以来，计算机经历了迅速的发展，获得了广泛的应用，极大地改变着人们的工作、学习和生活方式，并对国民经济的发展和科学技术的进步产生了巨大的推动作用。它的广泛应用是信息时代到来的主要标志。

计算机按其性能、体积和价格分为巨型机、大型机、中型机、小型机和微型机五类。微型计算机属于第四代电子计算机产品，即大规模及超大规模集成电路计算机，是电路技术不断发展、芯片集成度不断提高的产物。它诞生于20世纪70年代，由于它体积小，性价比高，因此广泛地应用在各行各业和家庭中，大大推广和普及了计算机及其技术的应用，加速了信息化社会的进程。

1.1 计算机的数制及其转换

计算机的最基本功能是进行数据的计算和加工处理。计算机中的数是以器件的两个不同物理状态来表示的，一个具有两种不同的稳定状态且能相互转换的器件即可用来表示一位二进制数。计算机中采用的就是二进制数字系统。凡是需要由计算机处理的各种信息，无论是文本、字符、图形，还是声音、图像，在输入计算机内部时，都必须以二进制编码(不是数)来表示，以方便存储、传送和处理。

计算机内部的信息分为两大类：控制信息和数据信息。控制信息是一系列控制命令，用于控制计算机进行相应的操作；数据信息是计算机操作的对象，通常又可分为数值数据和非数值数据。数值数据用于表示数量的大小，有确定的数值；非数值数据不表示数量的大小，而表示字符、汉字、逻辑数据等信息。

计算机所要处理的信息都要用“0”和“1”两个基本符号(即基2码)来编码表示，这是因为

以下三个原因:

(1) 基 2 码在物理上最易实现。例如, 可用“1”和“0”分别表示高、低两个电位, 或表示脉冲的有无或正负极性, 其可靠性较高。

(2) 采用基 2 码表示二进制数, 其编码、加减等运算规则简单。

(3) 基 2 码的两个符号“1”和“0”正好可用来表示逻辑数据的“真”与“假”, 可使计算机的逻辑运算简单方便。

因此, 计算机内部的各种信息都是用二进制编码来表示的。

1.1.1 数与数制

1. 数制的基本概念

人们在长期的生产实践和日常生活中创造了应用各种数字符号表示事物数量的方法, 这些数的表示方法称为数制。数制是以表示数值所用的数字符号的个数来命名的, 数制有多种形式, 例如人们最常用到的十进制。生活中也常常用到其它进制, 如十二进制、十六进制、六十进制, 以及计算机中使用的二进制等。

进位计数制, 又称位置计数制, 是用一组固定的数字符号和统一的进位规则进行计数的科学方法, 简称进位制。凡是按进位的方法计数的数制都是进位计数制, 它有一个规则, 就是 N 进制必须逢 N 进一。进位制中还有数码、基数、权三个常用术语。

数码: 数制中表示数值大小的不同数字符号。例如, 十进制中有 0、1、2、3、4、5、6、7、8、9 共 10 个数码。

基数: 也称基, 是指某进位计数制中每个数位(数字位置)上允许选用的基本数码的个数。如二进制数的基数为 2, N 进制数的基数为 N。基数体现了该数制中进位和借位的原则: 当在某一位数上计够一个基数时需要向上进 1; 反之, 从上位借 1 在下位当一个整基数来使用。

权: 数制中每一个数位所具有的基值(即此数位上所表示出来的最小数值)称为该位上的权, 也称位权。由于存在进位, 同一数字符号在不同的数位上所代表的数值是不同的。权一般以相应进位制的基数幂的形式来表示。例如, 十进制数 1234.56, 每个数位上的权是 10 的某次幂, 从左到右各位上的权分别为: 10^3 、 10^2 、 10^1 、 10^0 、 10^{-1} 、 10^{-2} 。每个数位上的数字所表示的数值是这个数字和该数位的权的乘积。因此, 对任意进位制数都可以写成按权展开的幂的多项式和的形式:

$$\begin{aligned} N &= \pm (a_{n-1} \times R^{n-1} + a_{n-2} \times R^{n-2} + \cdots + a_1 \times R^1 + a_0 \times R^0 + a_{-1} \times R^{-1} + \cdots + a_{-m} \times R^{-m}) \\ &= \pm \sum_{i=-m}^{n-1} a_i \times R^i \end{aligned} \quad (1.1)$$

式(1.1)中, i 是数位; n 和 m 为正整数, n 表示小数点左边的位数, m 表示小数点右边的位数; a_i 是任意进位制数 N 的第 i 位的数码; R 为基数, R^i 为第 i 位的权。

总的来看, 各种进位计数制均有以下几个主要特点:

- (1) 每种计数制有一个确定的基数 R , 其数码个数等于基数, 最大数码比基数小 1。
- (2) 每个数位上的数码乘以该数位的权就得到该数位上的数字所表示的数值。
- (3) 低位向高位的进位规则是“逢基数 R 进一”。在混合小数中, 小数点右移一位相当于乘以 R ; 反之, 相当于除以 R 。

为了区别各种计数制的数据, 常采用下述两种方法来表达:

(1) 在数字后面加相应的英文字母来表示该数的数制。如：十进制数用 D(Decimal) 表示, 通常计算机操作中默认使用的是十进制, 所以十进制数的后缀可以省略; 二进制数用 B(Binary) 表示, 如 101110B; 八进制数用 O(Octal) 表示, 为了不和 0 混淆, 也可以用 Q 来表示八进制数, 如 346Q; 十六进制数用 H(Hexadecimal) 表示, 如 19H。

(2) 在括号外边加数字下标, 这种表示方法比较直观。例如二进制数 10101 可以写成 $(10101)_2$ 。

2. 几种常用的进位计数制

计算机中采用的是二进制计数制。但是, 由于书写、键入、读出二进制数时极易出差错, 而微机的字长又都是 4 的整数倍数(分别为 4 位、8 位、16 位、32 位和 64 位等), 考虑到 $2^3=8$ 、 $2^4=16$, 因此在编程时, 为了使书写和阅读既方便又不易出错, 还常常采用八进制计数和十六进制计数。此外, 人们对十进制计数最为熟悉, 因此输入和输出计算机的数经常使用十进制数来表示。这样各种进位计数制之间就存在着一种对应转换关系。

(1) 十进制数(Decimal Notation)。

人们习惯使用的十进制数有以下特点:

① 10 个数码, 即 0、1、2、3、4、5、6、7、8 和 9。十进制数的基数是 10, 可以用 0~9 十个数字和一个小数点符号来表示任意十进制数。

② 每个数码表示的值不仅取决于数码本身, 还取决于它所处的位置。相同的数码在不同位置的权不同, 所表示的数值不同。十进制数各位的权是 10 的整数次幂。例如, 个位的权为 10^0 , 十位的权为 10^1 , 百位的权为 10^2 等。

③ 遵从“逢十进一, 借一当十”的规则。

例 1.1 $333.33 = 333.33D = (333.33)_{10}$

$$= 3 \times 10^2 + 3 \times 10^1 + 3 \times 10^0 + 3 \times 10^{-1} + 3 \times 10^{-2}$$

(2) 二进制数(Binary Notation)。

二进制数有以下特点:

① 2 个数码, 即 0 和 1, 二进制数的基数是 2。

② 二进制数各数位上的权是 2 的整数次幂。小数点左边的权是 2 的正次幂, 小数点右边的权是 2 的负次幂。二进制数的值可以用它的按权展开式表示。

③ 遵从“逢二进一, 借一当二”的规则。

例 1.2 $1011.01B = (1011.01)_2$

$$= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

另外, 二进制数还具有一些独特的性质, 这些性质奠定了当代计算机的设计基础。这些性质是:

① 具有两个不同的数码, 很容易在自然界中找到两个不同的稳定状态来表示。

② 小数点向右移一位, 数值增大一倍; 反之, 即小数点向左移一位, 数值减小一半。

③ 对于二进制的整数而言, 若其最低位为 1, 则值为奇数; 最低位为 0, 则值为偶数。

④ 二进制数的运算规则简单。如: $0+0=0$; $0+1=1$; $1+0=1$; $1+1=0$ (此时向上进位 1)。

可见, 两个一位的二进制数相加运算, 其本位和的运算正好是二值异或逻辑运算的关系, 进位则是二值与逻辑运算的关系。所以, 在计算机中, 二进制数的运算是用逻辑运算来实现的。

(3) 八进制数(Octal Notation)。

八进制数有以下特点:

① 8 个数码, 即 0、1、2、3、4、5、6 和 7, 八进制数的基数是 8。

② 八进制数各数位上的权是 8 的整数次幂。八进制数的值也可以用它的按权展开式来表示。

③ 遵从“逢八进一, 借一当八”的规则。

例 1.3 $248.15_{\text{Q}} = (248.15)_8 = 2 \times 8^2 + 4 \times 8^1 + 8 \times 8^0 + 1 \times 8^{-1} + 5 \times 8^{-2}$

(4) 十六进制数(Hexadecimal Notation)。

十六进制数有以下特点:

① 16 个数码, 即 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E 和 F, 其中 A~F 表示 10~15 这六个数, 十六进制数的基数是 16。

② 十六进制数各数位上的权是 16 的整数次幂。十六进制数的值也可以用它的按权展开式来表示。

③ 遵从“逢十六进一, 借一当十六”的规则。

例 1.4 $69\text{C.B5H} = (69\text{C.B5})_{16} = 6 \times 16^2 + 9 \times 16^1 + 12 \times 16^0 + 11 \times 16^{-1} + 5 \times 16^{-2}$

注意: 在实际表示时, 一个十六进制数如果最高位数字为字母(A~F), 则在字母前面必须加一个 0, 以便与指令名、变量名、数据等相区别, 如 0E0H。

二进制数、八进制数和十六进制数之间存在特殊的关系, 即一位八进制数恰好可以用三位二进制数来表示; 一位十六进制数恰好可以用四位二进制数来表示, 即八进制数和十六进制数可作为二进制数的缩写形式, 且它们之间的关系是唯一的。

计算机中常用的二进制数、八进制数、十进制数和十六进制数之间的对应关系如表 1.1 所示。

表 1.1 二进制、八进制、十进制和十六进制的对应关系

十进制数	二进制数	八进制数	十六进制数
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11

1.1.2 不同数制间的转换

由于人们习惯用十进制计数,在研究问题或讨论解题的过程时,总是用十进制来考虑和书写的。当考虑成熟后,要把问题变成计算机能够“看得懂”的形式时,就需把问题中的所有十进制数转换成二进制数,这就需要用到“十进制数转换成二进制数的方法”。在计算机运算得到二进制数的结果时,又需要用到“二进制数转换为十进制数的方法”,才能把运算结果用十进制形式显示出来。同样,有时我们也需要实现十进制数和其它进位制数之间的转换以及二进制数和八进制数、十六进制数之间的转换。所有这些不同进制的数码之间的转换都叫做数制转换。

1. 二进制数与十进制数之间的转换

1) 二进制数转换成十进制数

将一个二进制数转换成十进制数十分简单,只要将二进制数的每一位(0或1)分别乘以它所对应的权,然后把各乘积项加起来就可以求得二进制数的十进制数值。简单地说就是“按权展开后相加求和”。

例 1.5 将二进制数 1101.101 转换为十进制数。

其转换过程如下:

$$\begin{aligned} 1101.101\text{B} &= (1101.101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 8 + 4 + 1 + 0.5 + 0.125 \\ &= 13.625 \end{aligned}$$

故

$$1101.101\text{B} = 13.625$$

2) 十进制数转换成二进制数

十进制数转换为任意非十进制数时,整数和纯小数的转换方法不同。一个既有整数部分又有小数部分的十进制数,则须对整数和小数两部分分别进行转换,然后再组合起来。

① 十进制整数转换为二进制整数。

十进制整数转换为二进制整数通常采用以下两种方法:

方法一: 减权定位法, 也称降幂法。具体做法是将十进制数作为被减数,依次同距它最近的二进制高位权值比较,若够减则减去该位权值,并使对应位数码为1,再将差值作为被减数往下比较;若不够减,对应位数码为0,然后越过该位与低一位权值比较,重复此过程,直到差为0或达到所需精度(对应小数转换时)为止。此种方法既可用于整数的转换,又可用于小数的转换。该方法也可用于十进制数和八进制数、十进制数和十六进制数的转换。

例 1.6 用减权定位法将 163.75 转换为二进制数。

$$128(2^7) < 163 < 256(2^8)$$

	减权比较	A_i	权
整数部分:	$163 - 128 = 35$	$A_7 = 1$	2^7
	$35 < 64$	$A_6 = 0$	2^6
	$35 - 32 = 3$	$A_5 = 1$	2^5
	$3 < 16$	$A_4 = 0$	2^4
	$3 < 8$	$A_3 = 0$	2^3

	$3 < 4$	$A_2 = 0$	2^2
	$3 - 2 = 1$	$A_1 = 1$	2^1
	$1 - 1 = 0$	$A_0 = 1$	2^0
小数部分:	$0.75 - 0.5 = 0.25$	$A_{-1} = 1$	2^{-1}
	$0.25 - 0.25 = 0$	$A_{-2} = 1$	2^{-2}
	$163.75 = 10100011.11\text{B}$		

方法二：十进制整数转换为二进制整数更多的采用“除 2 取余，逆序排列”法。具体做法是将十进制整数除以二进制的基数 2，得到一个商和一个余数；取其余数（必定是 0 或 1）作为相应二进制整数的最低位 A_0 ，然后继续用商除以 2，又得到一个商和一个余数；取其余数作为二进制整数的次低位 A_1 ，依次重复此过程，直到商为 0 为止。

注意：第一次得到的余数为二进制数的最低位，最后得到的余数为二进制数的最高位。依次从最高位到最低位写出，就是整数部分的二进制数。

② 十进制小数转换为二进制小数。

十进制小数转换成二进制小数采用“乘 2 取整，顺序排列”法。具体做法是用二进制基数 2 乘以十进制小数，可以得到积的整数和小数部分，将积的整数部分（当十进制小数大于 0.5 时整数位为 1，当该小数小于 0.5 时整数位为 0）取出，作为二进制小数的小数点后的第一位（小数部分的最高位） A_{-1} ，然后再用 2 乘以余下的小数部分，又得到一个积，再将积的整数部分取出作为小数部分的第二位 A_{-2} ，如此重复该过程，直到积中的小数部分为 0，或者达到所要求的精度为止（有乘不尽的可能）。然后把每次取出的整数部分（必定是 0 或 1）按先后顺序从左到右排列起来，就得到转换后的二进制小数。

注意：第一次取出的整数为二进制小数的最高位，最后一次所取得的整数为其最低位。

例 1.7 将十进制数 139.8125 转换成二进制数。

其转换过程如下：

整数部分的转换：

$139 \div 2 = 69$	……余数为 1，对应二进制位 $A_0 = 1$ （最低位）
$69 \div 2 = 34$	……余数为 1，对应二进制位 $A_1 = 1$
$34 \div 2 = 17$	……余数为 0，对应二进制位 $A_2 = 0$
$17 \div 2 = 8$	……余数为 1，对应二进制位 $A_3 = 1$
$8 \div 2 = 4$	……余数为 0，对应二进制位 $A_4 = 0$
$4 \div 2 = 2$	……余数为 0，对应二进制位 $A_5 = 0$
$2 \div 2 = 1$	……余数为 0，对应二进制位 $A_6 = 0$
$1 \div 2 = 0$	……余数为 1，对应二进制位 $A_7 = 1$ （最高位）

整数部分的转换结果： $(139)_{10} = (10001011)_2$

小数部分的转换：

$0.8125 \times 2 = 1.625$	……整数为 1，对应二进制位 $A_{-1} = 1$ （最高位）
$0.625 \times 2 = 1.25$	……整数为 1，对应二进制位 $A_{-2} = 1$
$0.25 \times 2 = 0.5$	……整数为 0，对应二进制位 $A_{-3} = 0$
$0.5 \times 2 = 1.0$	……整数为 1，对应二进制位 $A_{-4} = 1$ （最低位）

小数部分的转换结果：

$$(0.8125)_{10} = (0.1101)_2$$

最后转化结果为:

$$(139.8125)_{10} = (10001011.1101)_2 \text{ 或 } 139.8125D = 10001011.1101B$$

例 1.8 将十进制数 27.3 转换为二进制数, 要求其精度为 4%。

其转换过程如下:

整数部分的转换:

$$27 \div 2 = 13 \quad \cdots \cdots \text{余数为 1, 对应二进制位 } A_0 = 1 (\text{最低位})$$

$$13 \div 2 = 6 \quad \cdots \cdots \text{余数为 1, 对应二进制位 } A_1 = 1$$

$$6 \div 2 = 3 \quad \cdots \cdots \text{余数为 0, 对应二进制位 } A_2 = 0$$

$$3 \div 2 = 1 \quad \cdots \cdots \text{余数为 1, 对应二进制位 } A_3 = 1$$

$$1 \div 2 = 0 \quad \cdots \cdots \text{余数为 1, 对应二进制位 } A_4 = 1 (\text{最高位})$$

整数部分的转换结果: $(27)_{10} = (11011)_2$

小数部分的转换:

由于精度要求为 4%, 故应该令 $2^{-m} \leq 4\% = 5^{-2}$, 可得 $2^m \geq 5^2 = 25$, 则 $m \geq 5$, 即要求其误差不大于 2^{-5} 。

$$0.3 \times 2 = 0.6 \quad \cdots \cdots \text{整数为 0, 对应二进制位 } A_{-1} = 0 (\text{权为 } 2^{-1}, \text{最高位})$$

$$0.6 \times 2 = 1.2 \quad \cdots \cdots \text{整数为 1, 对应二进制位 } A_{-2} = 1$$

$$0.2 \times 2 = 0.4 \quad \cdots \cdots \text{整数为 0, 对应二进制位 } A_{-3} = 0$$

$$0.4 \times 2 = 0.8 \quad \cdots \cdots \text{整数为 0, 对应二进制位 } A_{-4} = 0$$

$$0.8 \times 2 = 1.6 \quad \cdots \cdots \text{整数为 1, 对应二进制位 } A_{-5} = 1 (\text{权为 } 2^{-5}, \text{最低位})$$

小数部分的转换结果: $(0.3)_{10} = (0.01001)_2$

最后转化结果为: $(27.3)_{10} = (11011.01001)_2$

2. 八进制数与十进制数之间的转换

1) 八进制数转换成十进制数

同二进制转换成十进制的方法相似, 即按权展开后相加求和。

例 1.9 将八进制数 126.14 转换为十进制数。

其转换过程如下:

$$\begin{aligned} 126.14Q &= (126.14)_8 = 1 \times 8^2 + 2 \times 8^1 + 6 \times 8^0 + 1 \times 8^{-1} + 4 \times 8^{-2} \\ &= 64 + 16 + 6 + 0.125 + 0.0625 \\ &= 86.1875 \end{aligned}$$

即

$$(126.14)_8 = 86.1875$$

2) 十进制整数转换成八进制整数

十进制整数转换为八进制整数采用“除 8 取余, 逆序排列”法。具体做法是将十进制整数除以八进制的基数 8, 得到一个商和一个余数; 取其余数 (必定是小于 8 的整数) 作为相应八进制整数的最低位 A_0 , 然后继续用商除以 8, 又得到一个商和一个余数; 取其余数作为八进制整数的次低位 A_1 , 依次重复此过程, 直到商为 0 为止。

注意: 第一次得到的余数为八进制数的最低位, 最后得到的余数为八进制数的最高位。依次从最高位到最低位写出, 就是整数部分的八进制数。

3) 十进制小数转换为八进制小数

十进制小数转换成八进制小数采用“乘 8 取整, 顺序排列”法。具体做法是用八进制基数

8 乘以十进制小数,可以得到积的整数和小数部分,将积的整数部分(必定是小于 8 的整数)取出,作为八进制小数部分的最高位 A_{-1} ,然后再用 8 乘以余下的小数部分,又得到一个积,再将积的整数部分取出作为八进制小数部分的第二位 A_{-2} ,如此重复该过程,直到积中的小数部分为 0,或者达到所要求的精度为止。然后把每次的整数部分按先后顺序从左到右排列起来,就得到转换后的八进制小数。

例 1.10 将十进制数 239.32 转换成八进制数(转换结果取 4 位小数)。

其转换过程如下:

整数部分的转换:

$$239 \div 8 = 29 \quad \cdots \cdots \text{余数为 } 7, \text{ 对应八进制位 } A_0 = 7 (\text{最低位})$$

$$29 \div 8 = 3 \quad \cdots \cdots \text{余数为 } 5, \text{ 对应八进制位 } A_1 = 5$$

$$3 \div 8 = 0 \quad \cdots \cdots \text{余数为 } 3, \text{ 对应八进制位 } A_2 = 3 (\text{最高位})$$

整数部分的转换结果: $(239)_{10} = (357)_8$

小数部分的转换:

$$0.32 \times 8 = 2.56 \quad \cdots \cdots \text{整数为 } 2, \text{ 对应八进制位 } A_{-1} = 2 (\text{最高位})$$

$$0.56 \times 8 = 4.48 \quad \cdots \cdots \text{整数为 } 4, \text{ 对应八进制位 } A_{-2} = 4$$

$$0.48 \times 8 = 3.84 \quad \cdots \cdots \text{整数为 } 3, \text{ 对应八进制位 } A_{-3} = 3$$

$$0.84 \times 8 = 6.72 \quad \cdots \cdots \text{整数为 } 6, \text{ 对应八进制位 } A_{-4} = 6 (\text{最低位})$$

小数部分的转换结果: $(0.32)_{10} \approx (0.2436)_8$

最后转化结果为: $(239.32)_{10} = (357.2436)_8$ 或 $239.32D = 357.2436Q$

3. 十六进制数与十进制数之间的转换

1) 十六进制数转换成十进制数

同二进制数转换成十进制数的方法相似,按权展开后相加求和。

例 1.11 将十六进制数 5F6.C8 转换为十进制数。

其转换过程如下:

$$\begin{aligned} 5F6.C8H &= (5F6.C8)_{16} = 5 \times 16^2 + 15 \times 16^1 + 6 \times 16^0 + 12 \times 16^{-1} + 8 \times 16^{-2} \\ &= 1280 + 240 + 6 + 0.75 + 0.03125 \\ &= 1526.78125 \end{aligned}$$

即

$$5F6.C8H = 1526.78125D$$

2) 十进制整数转换成十六进制整数

十进制整数转换为十六进制整数采用“除 16 取余,逆序排列”法。具体做法是将十进制整数除以十六进制的基数 16,得到一个商和一个余数;取其余数(必定是小于 F 的数)作为相应十六进制整数的最低位 A_0 ,然后继续用商除以 16,又得到一个商和一个余数;取其余数作为十六进制整数的次低位 A_1 ,依次重复此过程,直到商为 0 为止。

注意:第一次得到的余数为十六进制数的最低位,最后得到的余数为十六进制数的最高位。依次从最高位到最低位写出,就是整数部分的十六进制数。

3) 十进制小数转换为十六进制小数

十进制小数转换成十六进制小数采用“乘 16 取整,顺序排列”法。具体做法是用十六进制基数 16 乘以十进制小数,可以得到积的整数和小数部分,将积的整数部分(必定是小于 F 的数)取出,作为十六进制小数部分的最高位 A_{-1} ,然后再用 16 乘以余下的小数部分,又得到

一个积,再将积的整数部分取出作为十六进制小数部分的第二位 A_{-2} ,如此重复该过程,直到积中的小数部分为0,或者达到所要求的精度为止。然后把每次获得的整数部分按先后顺序从左到右排列起来,就得到转换后的十六进制小数。

例 1.12 将十进制数 58312.46 转换成十六进制数(转换结果取 4 位小数)。

其转换过程如下:

整数部分的转换:

$$58312 \div 16 = 3644 \quad \cdots \cdots \text{余数为 } 8, \text{ 对应十六进制位 } A_0 = 8 (\text{最低位})$$

$$3644 \div 16 = 227 \quad \cdots \cdots \text{余数为 } 12, \text{ 对应十六进制位 } A_1 = C$$

$$227 \div 16 = 14 \quad \cdots \cdots \text{余数为 } 3, \text{ 对应十六进制位 } A_2 = 3$$

$$14 \div 16 = 0 \quad \cdots \cdots \text{余数为 } 14, \text{ 对应十六进制位 } A_3 = E (\text{最高位})$$

整数部分的转换结果: $(58312)_{10} = (E3C8)_{16}$

小数部分的转换:

$$0.46 \times 16 = 7.36 \quad \cdots \cdots \text{整数为 } 7, \text{ 对应十六进制位 } A_{-1} = 7 (\text{最高位})$$

$$0.36 \times 16 = 5.76 \quad \cdots \cdots \text{整数为 } 5, \text{ 对应十六进制位 } A_{-2} = 5$$

$$0.76 \times 16 = 12.16 \quad \cdots \cdots \text{整数为 } 12, \text{ 对应十六进制位 } A_{-3} = C$$

$$0.16 \times 16 = 2.56 \quad \cdots \cdots \text{整数为 } 2, \text{ 对应十六进制位 } A_{-4} = 2 (\text{最低位})$$

小数部分的转换结果: $(0.46)_{10} \approx (0.75C2)_{16}$

最后转化结果为: $(58312.46)_{10} = (E3C8.75C2)_{16}$ 或 $58312.46D = 0E3C8.75C2H$

4. 二进制数、八进制数和十六进制数之间的转换

由于 $2^3 = 8$, $2^4 = 16$, 3 位二进制数可以有 8 个状态, 即 000~111, 正好是八进制, 而 4 位二进制数可以有 16 个状态, 即 0000~1111, 正好是十六进制。这说明每三位二进制数对应一位八进制数; 每四位二进制数对应一位十六进制数, 因此二进制数与八进制数和十六进制数之间的相互转换很容易实现。

1) 二进制数转换成八进制数

二进制数转换成八进制数的方法是: 将二进制数从小数点所在位开始, 向左每三位分成一组, 然后写出每一组的等值八进制数, 若小数点左侧的位数不是 3 的整数倍, 则在二进制数的最左侧补 0, 这样得到整数部分的八进制数; 向右也每三位分成一组, 然后写出每一组的等值八进制数, 若小数点右侧的位数不是 3 的整数倍, 则在二进制数的最右侧补 0, 得到小数部分的八进制数, 最后把整数部分和小数部分顺序排列起来就得到所要求的八进制数。

例 1.13 将二进制数 10100011.11B 转换成八进制数。

其转换过程如下:

先分组	010	100	011	.	110
	↓	↓	↓		↓
	2	4	3		6

所以, 转换的结果为:

$$10100011.11B = 243.6Q$$

2) 八进制数转换成二进制数

八进制数转换成二进制数的方法是: 将每一位八进制数分别转换成对应的三位二进制数, 按顺序排列后即八进制数所对应的二进制数。

例 1.14 将八进制数 146.703Q 转换成二进制数。

其转换过程如下：

1	4	6	.	7	0	3
↓	↓	↓		↓	↓	↓
001	100	110	.	111	000	011

所以，转换的结果为：

$$146.703Q = 1100110.111000011B$$

3) 二进制数转换成十六进制数

二进制数转换成十六进制数的方法是将二进制数从小数点所在位开始，向左每四位分成一组，然后写出每一组的等值十六进制数，若小数点左侧的位数不是4的整数倍，则在二进制数的最左侧补0，这样得到整数部分的十六进制数；向右也每四位分成一组，然后写出每一组的等值十六进制数，若小数点右侧的位数不是4的整数倍，则在二进制数的最右侧补0，得到小数部分的十六进制数，最后把整数部分和小数部分顺序排列起来就得到所要求的十六进制数。

例 1.15 将二进制数 10100011.111001B 转换成十六进制数。

其转换过程如下：

先分组	1010	0011	.	1110	0100
	↓	↓		↓	↓
	A	3	.	E	4

所以，转换的结果为：

$$10100011.111001B = 0A3.E4H$$

4) 十六进制数转换成二进制数

十六进制数转换成二进制数的方法是将每一位十六进制数分别转换成对应的四位二进制数，按顺序排列后即为十六进制数所对应的二进制数。

例 1.16 将十六进制数 75A.88H 转换成二进制数。

其转换过程如下：

7	5	A	.	8	8
↓	↓	↓		↓	↓
0111	0101	1010	.	1000	1000

所以，转换的结果为：

$$75A.88H = 11101011010.10001B$$

八进制数和十六进制数主要用来简化二进制数的书写。由于采用八进制数和十六进制数表示的二进制数较短，且便于记忆，所以在 PC 机中主要使用十六进制数表示和编码二进制数，所以必须十分熟悉二进制数与十六进制数的对应关系。

另外，十六进制数和八进制数之间也可以进行相互转换，一般可通过先将十六进制数(或八进制数)转换成二进制数，然后再将二进制数转换成八进制数(或十六进制数)。

若要将十进制数转换成八进制或十六进制数，除采用前面介绍过的方法外，还可先将其转换成二进制数，然后再分组，进而转换成八进制数或十六进制数。

5. 二进制数的算术运算

采用二进制实现各种算术与逻辑运算，是因为二进制数中每一位都只有 0 和 1 两个数。它们表示两种不同状态，所以其物理过程很容易实现，如它很好地对应着电位的高与低、电