

21
世纪

21世纪高职高专系列教材

微机原理与应用

中国机械工业教育协会 组编



机械工业出版社
China Machine Press

21世纪高职高专系列教材

微机原理与应用

中国机械工业教育协会 组编

主 编 天津大学 曹玉珍
副主编 天津中德职业技术学院 邓 蓓
参 编 彭城职业大学 吕俊怀
天津大学 刘 鸣 丁北生
天津中德职业技术学院 孙 锋
主 审 江苏理工大学 杨文显



机械工业出版社

本书是根据高等职业技术院校的教学要求编写的。在介绍计算机基础知识和微机基本结构的基础上,以 8086CPU 为核心,详细介绍其工作原理和指令系统,以及汇编语言程序设计方法和上机调试过程,并适当介绍 80286 以上至 Pentium CPU 的特点。在重实用的原则下,阐述了有关微机接口器件的原理及应用。本书每章都附有一定数量的复习思考题。通过本书的学习,使读者具备一定的微机应用系统的开发能力和汇编语言程序设计能力。

本书可作为高等职业院校、高等学校专科、职工大学、业余大学、函授大学、成人教育学院大专层次的非计算机专业学生学习微机原理与应用的教材,也可作为从事微机软硬件工作的工程技术人员的参考用书。

图书在版编目(CIP)数据

微机原理与应用/中国机械工业教育协会组编.—北京:机械工业出版社,2001,6

21 世纪高职高专系列教材

ISBN7-111-08403-9

I. 微… II. 中… III. 微型计算机—高等学校:技术学校—教材 IV. TP36

中国版本图书馆 CIP 数据核字(2001)第 029817 号

机械工业出版社(北京百万庄大街 22 号 邮政编码 100037)

责任编辑:余茂祚

责任印制:郭景龙

中国农业出版社印刷厂印刷 新华书店北京发行所发行

2001 年 10 月第 1 版第 2 次印刷

787mm×1092mm $\frac{1}{16}$ • 17.5 印张 • 432 千字

5 001-9 000 册

定价:26.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换
本社购书热线电话(010) 68993821、68326677-2527

13-570/04

21 世纪高职高专系列教材编委会名单

编委会主任 中国机械工业教育协会 郝广发

编委会副主任 (单位按笔画排)

山东工程学院 仪垂杰
大连理工大学 唐志宏
天津大学 周志刚
甘肃工业大学 路文江
江苏理工大学 杨继昌
成都航空职业技术学院 陈玉华

机械工业出版社 陈瑞藻 (常务)
沈阳工业大学 李荣德
河北工业大学 檀润华
武汉船舶职业技术学院 郭江平
金华职业技术学院 余党军

编委会委员 (单位按笔画排)

广东白云职业技术学院 谢瀚华
山东省职业技术教育师资培训中心 邹培明
上海电机技术高等专科学校 徐余法
天津中德职业技术学院 李大卫
天津理工学院职业技术学院 沙洪均
日照职业技术学院 李连业
北方交通大学职业技术学院 佟立本
辽宁工学院职业技术学院 李居参
包头职业技术学院 郑刚
北京科技大学职业技术学院 马德青
北京建设职工大学 常莲
北京海淀走读大学 成运花
江苏理工大学 吴向阳
合肥联合大学 杨久志

同济大学 孙章
机械工业出版社 李超群 余茂祚 (常务)
沈阳建筑工程学院 王宝金
佳木斯大学职业技术学院 王跃国
河北工业大学 范顺成
哈尔滨理工大学工业技术学院 线恒录
洛阳大学 吴锐
洛阳工学院职业技术学院 李德顺
南昌大学 肖玉梅
厦门大学 朱立秒
湖北工学院高等职业技术学院 吴振彪
彭城职业大学 陈嘉莉
燕山大学 刘德有

序

1999年6月中共中央国务院召开第三次全国教育工作会议,作出了“关于深化教育改革,全面推进素质教育的决定”的重大决策,强调教育在综合国力的形成中处于基础地位,坚持实施科教兴国的战略。决定中明确提出要大力发展高等职业教育,培养一大批具有必备的理论知识和较强的实践能力,适应生产、建设、管理、服务第一线急需的高等技术应用性专门人才。为此,教育部召开了关于加强高职高专教学工作会议,进一步明确了高职高专是以培养技术应用性专门人才为根本任务;以适应社会需要为目标;以培养技术应用能力为主线设计学生的知识、能力、素质结构和培养方案;以“应用”为主旨和特征来构建课程和教学内容体系;高职高专的专业设置要体现地区、行业经济和社会发展的需要,即用户的需求;教材可以“一纲多本”,形成有特色的高职高专教材系列。

“教书育人,教材先行”,教育离不开教材。为了贯彻中共中央国务院以及教育部关于高职高专人才培养目标及教材建设的总体要求,中国机械工业教育协会、机械工业出版社组织全国部分有高职高专教学经验的职业技术学院、普通高等学校编写了这套《21世纪高职高专系列教材》。教材首批80余本(书目附书后)已陆续出版发行。

本套教材是根据高中毕业3年制(总学时1600~1800)、兼顾2年制(总学时1100~1200)的高职高专教学计划需要编写的。在内容上突出了基础理论知识的应用和实践能力的培养。基础理论课以应用为目的,以必需、够用为度,以讲清概念、强化应用为重点;专业课加强了针对性和实用性,强化了实践教学。为了扩大使用面,在内容的取舍上也考虑到电大、职大、业大、函大等教育的教学、自学需要。

每类专业的教材在内容安排和体系上是有机联系、相互衔接的,但每本教材又有各自的独立性。因此各地区院校可根据自己的教学特点进行选择使用。

为了提高质量,真正编写出有显著特色的21世纪高职高专系列教材,组织编写队伍时,采取专门办高职的院校与办高职的普通高等院校相互协作编写并交叉审稿,以便实践教学和理论教学能相互渗透。

机械工业出版社是我国成立最早、规模最大的科技出版社之一,在教材编辑出版方面有雄厚的实力和丰富的经验,出版了一大批适用于全国研究生、大学本科、专科、中专、职工培训等各种层次的成套系列教材,在国内享有很高的声誉。我们相信这套教材也一定能成为具有我国特色的、适合21世纪高职高专教育特点的系列教材。

中国机械工业教育协会

前 言

本教材是在编者多年从事职业教育实践的基础上,根据高职高专教育的实际需要编写的。在编写过程中,以应用为目的,以必要、够用为度,以讲清概念、强化应用为重点。具体为:

1. 以目前国内的主流机型 IBM PC 微型计算机系统为背景,系统介绍了微机系统的构成和工作原理;以 8086 CPU 为核心,详细介绍其寻址方式和指令系统;并适当介绍 80286 以上至 Pentium CPU 的特点。

2. 由浅入深地介绍汇编语言程序设计方法和程序设计实例,每个设计实例均已调试通过,可实际运行。

3. 简要阐述了微机系统中的存储器、中断系统和输入输出系统,并介绍了有关微机接口器件的原理及在微机系统中的应用。

4. 以实用为目的,介绍了汇编语言程序的上机调试过程,并将“实验指导”作为专门的一章编入教材,对每个实验提出了实验要求且提供实验参考。

本教材第 1、2 章由天津中德职业技术学院邓蓓编写,第 3、6 章由天津大学曹玉珍编写,第 4、10 章由天津大学刘鸣编写,第 5 章由天津中德职业技术学院孙锋编写,第 7 章和附录由天津大学丁北生编写,第 8、9 章由彭城职业大学吕俊怀编写。全书由曹玉珍主编并统稿,承蒙江苏理工大学杨文显老师细心审阅。在编写过程中,得到天津大学教务处、职教学院的大力支持,在此表示衷心的感谢。

本书作为高职高专非计算机专业开设“微机原理”课程的教材时,参考学时为:课内 80 学时,实验 20 学时。使用时可根据各院校的教学特点、学时的多少,在内容上有所侧重或删减。

由于编者水平有限,书中难免有错误和不妥之处,恳请广大读者批评指正。

编 者

目 录

序	
前言	
第 1 章 计算机基础知识	1
1.1 计算机概述	1
1.1.1 计算机的历史	1
1.1.2 计算机的发展	1
1.1.3 微处理器的发展	2
1.2 计算机中的数和编码系统	2
1.2.1 进位计数制	2
1.2.2 计算机中常用的编码	5
1.2.3 计算机中带符号数的表示	7
1.3 计算机中常用术语	12
复习思考题	13
第 2 章 微型计算机结构	15
2.1 微型计算机的基本结构	15
2.2 Intel 8086/8088 CPU	16
2.2.1 8086/8088 CPU 的功能结构	16
2.2.2 8086/8088 CPU 的寄存器结构	18
2.2.3 8086/8088 CPU 的引脚和功能	20
2.2.4 存储器组织	30
复习思考题	34
第 3 章 指令系统	36
3.1 指令与指令系统	36
3.2 8086 指令系统的基本寻址方式	36
3.2.1 立即寻址方式	36
3.2.2 寄存器寻址方式	37
3.2.3 直接寻址方式	37
3.2.4 寄存器间接寻址方式	37
3.2.5 寄存器相对寻址方式	38
3.2.6 基址变址寻址方式	38
3.2.7 基址变址相对寻址方式	39
3.3 8086/8088 指令系统	39
3.3.1 数据传送指令	39
3.3.2 算术运算指令	45
3.3.3 位操作指令	52
3.3.4 串操作指令	56
3.3.5 控制传送指令	59
3.3.6 处理器控制类指令	64
复习思考题	65
第 4 章 汇编语言程序设计	68
4.1 汇编语言程序	68
4.1.1 汇编语言的基本概念	68
4.1.2 汇编语言源程序的格式	68
4.2 汇编语言的语句	69
4.2.1 指令语句	69
4.2.2 伪指令语句	73
4.2.3 宏指令语句	79
4.3 汇编语言程序的上机过程及调试	80
4.3.1 编辑汇编语言源程序	80
4.3.2 汇编源程序	81
4.3.3 连接程序	83
4.3.4 程序的执行	84
4.4 汇编语言程序设计的基本方法	84
4.4.1 程序设计的基本方法	84
4.4.2 顺序程序设计	84
4.4.3 分支程序设计	87
4.4.4 循环程序设计	89
4.4.5 子程序设计	92
4.4.6 DOS 系统功能调用	97
4.5 汇编语言程序设计应用	100
4.5.1 算术运算	100

4.5.2 代码转换	103	6.4.2 8237A 的工作方式	135
复习思考题	108	6.4.3 8237A 的内部寄存器和编程 ..	135
第 5 章 存储器	110	6.4.4 DMA 操作过程与时序	139
5.1 半导体存储器	110	6.4.5 PC 机中的 DMA 系统	140
5.1.1 存储器概述	110	6.5 微机中常用的几种总线标准 ..	142
5.1.2 半导体存储器分类与性能	111	6.5.1 PC 总线	142
5.2 随机存取存储器(RAM)	113	6.5.2 ISA (16 位) 总线	143
5.2.1 静态 RAM (SRAM)	113	6.5.3 局部总线	144
5.2.2 DRAM 芯片实例	114	复习思考题	145
5.2.3 几种新型 RAM 技术	116	第 7 章 中断	146
5.3 只读存储器 (ROM)	117	7.1 中断原理	146
5.4 存储器与 CPU 的连接	118	7.1.1 中断的定义与作用	146
5.4.1 连接中应考虑的问题	118	7.1.2 中断源	146
5.4.2 存储器容量扩展	118	7.1.3 中断源管理	147
5.4.3 存储器的地址选择	119	7.1.4 中断过程	148
5.4.4 8086 CPU 与主存储器的连接 ..	121	7.2 8086 CPU 中断系统	149
5.5 PC 机存储系统的层次结构和 对内存的管理	122	7.2.1 8086 的中断类型	149
5.5.1 PC 机存储系统的层次结构	122	7.2.2 8086 的中断矢量表	150
5.5.2 内存管理	123	7.3 可编程中断控制器 8259A ...	151
5.6 存储器的新技术展望	124	7.3.1 8259A 引脚及内部结构	151
复习思考题	125	7.3.2 8259A 中断触发方式和中断响 应过程	153
第 6 章 输入和输出系统	127	7.3.3 8259A 的编程	154
6.1 概述	127	7.4 IBM PC/XT 的中断结构	164
6.1.1 CPU 与外设间交换的信息	127	7.4.1 IBM PC/XT 的中断矢量	164
6.1.2 典型 I/O 接口形式	127	7.4.2 8259A 与 IBM PC/XT 系统总线 的连接	165
6.2 I/O 端口的编址方法	128	复习思考题	166
6.2.1 I/O 端口与内存统一编址	128	第 8 章 微机接口技术与应用	167
6.2.2 I/O 端口与内存独立编址	128	8.1 并行 I/O 接口	167
6.2.3 PC 机中 I/O 端口地址分配	128	8.1.1 并行接口的基本概念	167
6.3 I/O 控制方式	129	8.1.2 可编程并行接口芯片 —— 8255A	168
6.3.1 无条件传送方式	130	8.2 串行通信接口	179
6.3.2 查询传送方式	130	8.2.1 串行通信的基本概念	179
6.3.3 中断传送方式	132	8.2.2 可编程串行通信接口芯片 —— 8251A	182
6.3.4 直接存储器存取 (DMA) 方式 ..	132		
6.4 DMA 控制器 8237A	133		
6.4.1 8237A 的功能	133		

8.3 定时/计数技术	189	9.3.1 概述	234
8.3.1 定时/计数的基本概念	189	9.3.2 Pentium 微处理器的结构	234
8.3.2 可编程定时器/计数器 8253	189	9.3.3 Pentium 微处理器的内部 寄存器	236
8.4 A/D 和 D/A 接口	203	9.3.4 新一代微处理器的其他功能	237
8.4.1 模拟量和数字量	203	复习思考题	238
8.4.2 D/A 转换器及其接口	204	第 10 章 实验指导书	239
8.4.3 A/D 转换器及其接口	210	10.1 DEBUG 的使用	239
复习思考题	216	10.2 软件实验部分	242
第 9 章 高性能微处理器	218	10.3 硬件实验部分	244
9.1 80X86 系列微处理器	218	附录 A ASCII 字符表	255
9.1.1 80X86 系列微处理器概述	218	附录 B 8086 指令表	255
9.1.2 80X86 的寄存器概述	218	附录 C DOS 系统功能调用表	260
9.2 80486 微处理器简介	219	附录 D BIOS 中断功能调用表	264
9.2.1 80486 微处理器概述	219	附录 E DOS 功能调用表	266
9.2.2 80486 微处理器的功能结构	219	附录 F 汇编出错代码注释	267
9.2.3 80486 微处理器的引脚信号	226	参考文献	270
9.2.4 80486 微处理器的工作方式	229		
9.3 Pentium 微处理器	234		

第 1 章 计算机基础知识

1.1 计算机概述

1.1.1 计算机的历史

计算机是一种现代的信息处理工具，其主要特性是自动化、通用化。作为一种解决问题的工具，电子计算机可以解析自然界中各种现象，大到宇宙天体、人类社会，小到企业、事业家庭中各种问题。

虽然第一台电子数字计算机问世至今，只有短短 50 多年历史，但计算机的渊源可以追溯到 2000 年以前。那时，我国人民已发明了算盘，用它进行计算，这是人类借助于工具进行数字计算的开端。人们认为，现代计算机的前身是帕斯卡在 1642 年建造的第一台机械式的计算机，后来，巴贝奇在 1823 年建造了机械式的数字微分机，这些都是在电子计算机诞生前，人类创造计算工具的重要实践。

此外，早期奠定计算机理论的工作有：1936 年图灵提出的“图灵机”，构成了现代计算机的理论基础。按照这个理论，一台通用的图灵机只要供给适当的程序，就可以执行任何预定的任务。1946 年，冯·诺依曼又确立了存储程序计算机理论，现在实用的计算机都属于冯·诺依曼型计算机。至此，现代计算机的理论体系基本上确立。

1.1.2 计算机的发展

第一台成功的高速电子计算机是 1946 年在宾夕法尼亚大学建成的 ENIAC，它用 18000 个真空管构成，占地近 200m²，从此，计算机开始了它的发展历程。1947 年肖克莱等人发明了晶体管，第一代真空管计算机很快过渡到了第二代晶体管计算机，计算机的性能和可靠性大大提高。早期的计算机都是大型机，以 IBM 等公司的产品为代表，例如 IBM360 机在当时有很大的影响。我们把这一时期称为大型机时期。

1960 年左右集成电路问世，随之出现了第三代的集成电路计算机。随着集成电路技术的进步，计算机可以做得更小，更廉价，应用于更多的领域。在 20 世纪 70 年代，小型机得到了迅速发展，其代表是 Digital 公司的 VAX 机。这个时期我们称之为小型机时期。

从 1970 年出现第一个微处理器 4004 开始，微处理器在集成电路技术迅速发展的基础上，性能得到很大提高。集成电路的复杂性（性能）大约每隔 18 个月提高一倍，这一描述被称为摩尔定律，估计这个定律在今后十年左右的时间里仍可适用。到了 20 世纪 80 年代，超大规模集成电路（VLSI）问世，出现了第四代即 VLSI 计算机。这时，可以用微处理器构成低价格、高性能的计算机。1981 年 IBM 公司推出了划时代的、被称作“个人计算机”的 IBM PC 机，它标志着 PC 时期的开始。此后，计算机应用方式产生了很大的变化。在大型机和小型机时期，很多用户通过终端共用一台主机，称为分时共享方式。现在，各个用户都可以用各自的 PC 机独立工作，并通过局域网共享服务器上的文件，随后又发展到共享服务器上的应用系统，这种工作方式被称为“客户机/服务器”方式，因此这时期也被称为客户机/服务器时期或 PC/服务器时期。

1.1.3 微处理器的发展

1971 年 Intel 公司首先推出了第一个 4 位微处理器芯片 (Intel 4004), 集成了 2300 个晶体管。同年, 第一台使用 4004 芯片的微型计算机 (MCS-4 系统) 宣告诞生。其后, 大量 8 位微处理器迅速推向市场。在这一时期, 微型机的操作系统、高级语言、工具软件和应用软件日趋完善。

1981 年 IBM 公司推出的 IBM PC 机是微型机发展的一个里程碑。IBM 公司采用了 Intel 公司的 8088 微处理器和 Microsoft (微软) 公司的 MS-DOS 操作系统, 并公布了 IBM PC 的总线设计, 这些举措促进了兼容机的发展和微机的规模生产, 形成了事实上的标准。以后 Intel 公司又陆续推出了 80286、80386 微处理器芯片取代 8086/8088。与此同时, AMD 公司和 Cyrix 等公司也生产 Intel 80×86 系列的兼容产品而成为 Intel 公司的主要竞争对手。兼容微处理器产品以更高的性能价格比赢得了一定的市场份额。

Intel 公司的 32 位微处理器产品始于 80386, 并于 1993 年推出 Pentium 微处理器, 接着 Pentium MMX (带多媒体指令的 Pentium 微处理器)、Pentium Pro、Pentium II 和 Pentium III 相继问世。Pentium III 的主频可达到 450MHz 以上, 增强了浮点运算、并行处理、图像处理和连接因特网的功能。最新推出的 Pentium IV 微处理器主频已达 1.7GHz。

由于微处理器价格低廉, 用途广泛, 现在它几乎已达到了“无处不在”的地步。许多家用电器, 例如电冰箱和洗衣机中, 也采用微处理器芯片进行控制。大量的系统软件和应用软件被开发出来, 在功能越来越强大的同时, 使用方法越来越简单。未来微处理器应用量最大的领域之一将是近几年来在国际、国内推广的 CPU 卡市场。CPU 卡是集成电路卡 (IC 卡) 中性能最先进、工作最可靠的卡, 在名片大小的卡上嵌入了一块微处理器芯片。CPU 卡可用作识别身份的身份证、驾驶证和护照等; 也可作为支付凭证和储蓄卡、信用卡、电子钱包以及电子病历等。在 21 世纪, CPU 卡将会成为微处理器发展最快的应用领域。随着网络应用的推广和因特网 (Internet) 的普及, 微处理器与通信功能的结合是一个重要的发展趋势。

1.2 计算机中的数和编码系统

1.2.1 进位计数制

1. 进位计数制的基本概念 数制是人们利用符号来计数的规则和方法。日常生活中, 人们通常使用十进制计数。十进制的特点是“逢十进一”。十进制计数系统中, 有十个不同的数字符号 0~9。电子计算机工作时使用二进制, 对它进行程序设计时则使用二进制、十进制、八进制和十六进制。

我们知道, 在一个十进制数中, 同一个数字符号处在不同位置上所代表的值是不同的, 例如数字 5 在十位数位置上时表示 50, 在百位数位置上时表示 500, 而在小数点后第 1 位上则表示 0.5。同一个数字符号, 不管它在哪一个十进制数中, 只要在相同位置上其值是相同的, 如 256 与 1256 中的数字 5 都是在十位数位置上, 它们都表示 50。通常我们称某个固定位置上的计数单位为“位权”或“权”, 每一位数码与该位“位权”的乘积表示了该位数值的大小。十进制计数中的 10 称为基数。“位权”和“基数”是进位计数制的两个要素。

日常生活中, 除了采用十进制数外, 有时也采用别的进制计数。例如, 计算时间采用六十进制, 1h 为 60min, 1min 为 60s, 其特点为“逢六十进一”。

在计算机内部, 一切信息 (包括数值、字符、指挥计算机动作的指令等) 的存储、处理

与传送均采用二进制的形式。一个二进制数在计算机内部是以电子器件的物理状态来表示的，这些器件具有两种不同的稳定状态（低电平表示 0，高电平表示 1），并且这两种稳定状态之间能够互相转换。由于八进制、十六进制与二进制间有着非常简单的对应关系，而且位数变少，在阅读与书写时常常采用八进制或十六进制。表 1-1 和表 1-2 给出了常用的四种计数法的表示及其相互关系。

表 1-1 四种计数法的表示

计 数 法	二 进 制	八 进 制	十 进 制	十 六 进 制
进位规则	逢二进一	逢八进一	逢十进一	逢十六进一
基数 R	2	8	10	16
所用符号	0, 1	0, 1, 2, ..., 7	0, 1, 2, ..., 9	0, 1, 2, ..., 9, A, B, ..., F
权	2^i	8^i	10^i	16^i
数制标识	B	Q	D	H

表 1-2 四种计数法表示数的对应关系

十进制数	二进制数	八进制数	十六进制数	十进制数	二进制数	八进制数	十六进制数
0	0	0	0	9	1001	11	9
1	1	1	1	10	1010	12	A
2	10	2	2	11	1011	13	B
3	11	3	3	12	1100	14	C
4	100	4	4	13	1101	15	D
5	101	5	5	14	1110	16	E
6	110	6	6	15	1111	17	F
7	111	7	7	16	10000	20	10
8	1000	10	8				

2. 四种不同进制数的相互转换

(1) 任意进制数的表示 任意一个 n 位整数， m 位小数的 R 进制数都可用它的按权展

开公式表示：
$$S = \sum_{i=-m}^{n-1} r_i R^i$$

即：
$$S = r_{n-1}R^{n-1} + r_{n-2}R^{n-2} + r_{n-3}R^{n-3} + \cdots + r_1R^1 + r_0R^0 + r_{-1}R^{-1} + \cdots + r_{-m}R^{-m}$$

其中， r_i 为计数制中任一个数字， R 为基数。

例如：十进制数 309.84 可根据按权展开式写成：

$$3 \times 10^2 + 0 \times 10^1 + 9 \times 10^0 + 8 \times 10^{-1} + 4 \times 10^{-2}$$

对于八进制数 $(406)_8 = 4 \times 8^2 + 6 \times 8^0$

二进制数 $(110101)_2 = 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^2 + 1 \times 2^0$

十六进制数 $(AC7.B)_{16} = 10 \times 16^2 + 12 \times 16^1 + 7 \times 16^0 + 11 \times 16^{-1}$

按权展开式的值就是该数转换为十进制的等价值。

(2) 二进制与十进制数的相互转换

1) 二进制数转换为十进制数：可直接按权展开。

例1 请把二进制数 $(11011)_2$ 转化为十进制数。

解: $(11011)_2 = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0 = 16 + 8 + 2 + 1 = 27$

例2 请把二进制数 $(111.011)_2$ 转化为十进制数。

解: $(111.011)_2 = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-2} + 1 \times 2^{-3} = 4 + 2 + 1 + 0.25 + 0.125$
 $= 7.375$

2) 十进制数转换为二进制数: 方法是: 整数除以2取余, 小数乘以2取整。

例3 将十进制数 37.375 转换为二进制数, 其方法如下。

解:
$$\begin{array}{cccccccccccc} 0 & \leftarrow & 1 & \leftarrow & 2 & \leftarrow & 4 & \leftarrow & 9 & \leftarrow & 18 & \leftarrow & 37 & .375 & \rightarrow & .75 & \rightarrow & .5 & \rightarrow & 0 \\ & & \downarrow \div 2 & & \downarrow \div 2 & & \downarrow \div 2 & & \downarrow \div 2 & & \downarrow \div 2 & & \downarrow \div 2 & & \downarrow \times 2 & & \downarrow \times 2 & & \downarrow \times 2 & & \\ & & 1 & & 0 & & 0 & & 1 & & 0 & & 1 & & 0 & & 1 & & 1 & & \end{array}$$

故: $37.375 = (100101.011)_2$

另外, 也可采用记权值的方法转换为二进制数, 例如 $(86.625)_{10}$ 转换为二进制数。我们知道, 二进制数权值依次为:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	二进制权值
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	
128	64	32	16	8	4	2	1	0.5	0.25	0.125	十进制数

我们发现这样的规律, 以 2^0 为基准向左每一位依次 $\times 2$, 向右每一位依次 $\div 2$ 就得到上面表中的十进制值。因此, 86 可看成由 $64+16+4+2$ 组成, 0.625 可看成 $0.5+0.125$, 那么在相应的权值下面写上 1, 其余写 0, 就是它等值的二进制数。

64	32	16	8	4	2	1	0.5	0.25	0.125	0.0625
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
1	0	1	0	1	1	0	1	0	1	0

故: $(86.625)_{10} = (1010110.101)_2$

(3) 二进制数与八进制数的相互转换

1) 二进制数转换成八进制数: 方法是: 将二进制数自小数点开始分别向左、向右划段, 每 3 位划为一段, 不足 3 位者, 用 0 补满, 每段写成 1 位八进制数。

例4 将 $(1101100.0111011)_2$ 转换为八进制数。

解:
$$\begin{array}{ccccccc} \underline{001} & 101 & 100 & . & 011 & 101 & \underline{100} \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow \\ 1 & 5 & 4 & & 3 & 5 & 4 \end{array} \quad (\text{带下划线的数字是分组后添加的})$$

故: $(1101100.0111011)_2 = (154.354)_8$

2) 八进制数转换成二进制数: 方法是: 将八进制数整数部分和小数部分的数字逐个用对应的 3 位二进制数替代即可。

例5 将 $(635.05)_8$ 转换为二进制数。

解:
$$\begin{array}{cccccc} 6 & 3 & 5 & . & 0 & 5 \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ 110 & 011 & 101 & . & 000 & 101 \end{array}$$

故: $(635.05)_8 = (110011101.000101)_2$

例6 将 $(742.413)_8$ 转换为二进制数。

解: 7 4 2 . 4 1 3
 ↓ ↓ ↓ ↓ ↓ ↓
 111 100 010 . 100 001 011

故: $(742.413)_8 = (111100010.100001011)_2$

(4) 二进制数与十六进制数的相互转换

1) 二进制数转换为十六进制数: 方法是: 从小数点开始, 分别向左、向右每 4 位二进制数划为一段, 不足 4 位者填 0 补足。每段二进制数用 1 位十六进制数替代。

例 7 将 $(1011011100011.0011011)_2$ 转换成十六进制数。

解: 0001 0110 1110 0011 . 0011 0110 (带下划线的数字是分组后添加的)
 ↓ ↓ ↓ ↓ ↓ ↓
 1 6 E 3 . 3 6

故: $(1011011100011.0011011)_2 = (16E3.36)_{16}$

例 8 将 $(11111011001.01)_2$ 转换成十六进制数。

解: 0111 1101 1001 . 0100 (带下划线的数字是分组后添加的)
 ↓ ↓ ↓ ↓
 7 D 9 4

故: $(11111011001.01)_2 = (7D9.4)_{16}$

2) 十六进制数转换成二进制数: 方法是: 将十六进制数的整数部分和小数部分用相应的 4 位二进制数替代即可。

例 9 将 $(AE7.D2)_{16}$ 转换成二进制数。

解: A E 7 . D 2
 ↓ ↓ ↓ ↓ ↓
 1010 1110 0111 . 1101 0010

故: $(AE7.D2)_{16} = (101011100111.1101001)_2$

例 10 $(2C5.21F8)_{16}$ 转换成二进制数。

解: 2 C 5 . 2 1 F 8
 ↓ ↓ ↓ ↓ ↓ ↓ ↓
 0010 1100 0101 0010 0001 1111 1000

故: $(2C5.21F8)_{16} = (1011000101.0010000111111)_2$

为了便于区分不同数制所表示的数, 规定在数字尾部用 B 表示二进制数, 用 Q 表示八进制数, 用 D 表示十进制数(也可不加), 用 H 表示十六进制数。如 78H、756Q、853D、1101101B 分别表示十六进制、八进制、十进制和二进制数。另外, 规定当十六进制数以字母开头时, 为了避免与其他符号混淆, 通常在前面加一个数 0, 如十六进制数 A9H, 应写成 0A9H。

1.2.2 计算机中常用的编码

1. 十进制数的二进制编码 在现实生活中, 人们习惯用十进制数, 计算问题的原始数据大多是十进制数。此时, 可以有两种选择: 第一种选择是用上面介绍的方法把十进制数转换成二进制数, 另一种选择是用二进制数表示十进制, 保留各数位之间“逢十进一”的关系, 这就是二—十进制编码或称 BCD 码(Binary Coded Decimal)。这种编码将一位十进制数用 4 位二进制数表示, 通常以 8421 为权进行编制。它有十个不同的数字符号, 按“逢十进一”原

则进位。十进制与 8421 BCD 码的对应关系列于表 1-3 中。

表 1-3 十进制与 BCD 码对照表

十 进 制 数	BCD 码	十 进 制 数	BCD 码
0	0000	8	1000
1	0001	9	1001
2	0010	10	00010000
3	0011	11	00010001
4	0100	12	00010010
5	0101	13	00010011
6	0110	14	00010100
7	0111	15	00010101

例 11 将 $(867.51)_{10}$ 转换成 BCD 码。

解: $(867.54)_{10} = (1000\ 0110\ 0111.0101\ 0100)_{BCD}$

例 12 将 $(1101010110.01010011011)_{BCD}$ 转换成十进制数。

解: $(0011\ 0101\ 0110.0101\ 0011\ 0110)_{BCD} = (356.536)_{10}$

需要进行 BCD 码与其他进制数之间转换时, 可将 BCD 码先转换成十进制数, 再将十进制数转换成其他进制数, 反之亦然。例如: 1FDH 转换成 BCD 码, 先将 1FDH 转换成十进制数再转换为 BCD 码。

$(1FD)_{16} = (509)_{10} = (0101\ 0000\ 1001)_{BCD}$

BCD 码与十进制数转换方便, 易于阅读书写, 但是用 BCD 码表示的十进制数的位数要比纯二进制数表示的位数长, 运算规则复杂, 会造成电路复杂, 且影响运算速度。

2. 字符的编码 计算机中的每个字符均按某种规则, 用一组二进制编码表示。目前微机中应用最普遍的是美国标准信息交换码 (简称 ASCII 码, American Standard Code for Information Interchange), EBCDIC 码 (Extended Binary Coded Decimal Interchange Code) 和对汉字进行编码的 GB-2312 国标码。计算机能够识别如此多的字符和汉字, 就是因为它们有各自的编码。下面分别介绍它们的编码。

(1) ASCII 码 ASCII 码用 7 位二进制码对字符进行编码, 如附录 A 所示。ASCII 字符集共有 128 种常用字符, 其中数字字符 0~9, 大小写英文字母, 一些在算式中与语句、文本中常用的符号 (如四则运算符、括号、标点符号、特殊符号等), 还有一些控制字符。这些字符大致能满足各种编程语言、西文文字、常见控制命令等的需要。

每个 ASCII 码字符用七位编码, 最高位用 0 填充, 或者加一位奇偶校验位构成一个满字节。存储器中以字节作为基本的编址单位, 正好存放一个字符的 ASCII 码。

通用键盘的大部分键与最常用的字符相对应。在键盘上输入时, 系统软件用扫描法判明所按键的行列位置, 组织成扫描码 (表示该键在键盘上所在位置的编码), 再通过查表或其他方法, 最终转换成 ASCII 码, 存入存储器中供处理。计算机将结果输出时, 把 ASCII 码表示的字符送往显示器或打印机, 再通过其中的字符发生器转换为该字符的点阵图形。

128 个 ASCII 码字符分为可显示字符和非显示字符两类。可显示字符是指编码从 20H 到 7EH 的 95 个代码。它们可以从键盘终端上输入, 可在屏幕终端上显示, 也可在打印机上打印出来。非显示字符是编码从 00H 到 1FH 的 32 个代码, 还有编码为 3FH 的字符共 33 个,

它们主要用来控制输入输出设备。例如回车 (0DH) 使显示器的光标回到一行的首部, 换行 (0AH) 使显示器光标移到下一行。连续输出回车和换行就结束本行输出, 光标移到下一行首部, 开始新一行的输出。

(2) 汉字的编码 由于汉字的数量多 (常用汉字就有几千个), 它的编码相对要复杂些, 编码后的二进制位数也较多。汉字的编码有机内码和机外码两类。机内码是汉字的标识码, 机外码则用于汉字的输入, 以满足不同的需要。下面分别介绍汉字的国标码、机内码和机外码。

1) 国标码: 为了在信息交换中有个通用标准, 我国在 1981 年公布了国家标准 GB2312—1980《信息交换用汉字编码字符基本集》, 简称国标码。在这一标准中, 每个汉字用两个字节 (各使用七位二进制) 表示, 第一个字节表明字符位于哪一区, 第二字节表明该字符在本区内的哪一位。基本字符集共有 94 区, 每区有 94 位。另外, 该标准又按使用频率, 把常用汉字分为一级汉字 (3755 个) 和二级汉字 (3008 个)。一级汉字按拼音顺序排列, 占据 16~55 区; 二级汉字按部首顺序排列, 占据 56~87 区。1~15 区用来编排西文字母、数字、图形符号, 以及用户自行定义的专用符号。

2) 机内码: 机内码是计算机系统内部用来表示汉字的编码。计算机系统所采用的机内码以 GB2312—1980 码为基础, 为使该码与 ASCII 码有所区别, 将汉字国标码每个字节的最高位置 1, 作为该汉字的机内码。例如, 阿字的国标码是 3022H, 其机内码为 B0A2H。

3) 机外码: 目前常用的机外码主要有区位码、国标码、首尾码、拼音码、五笔字型等。

1.2.3 计算机中带符号数的表示

计算机内的数分为无符号数和带符号数。无符号数只能表示正数, 带符号数可表示正数和负数。计算机中, 为便于识别, 需将带符号数的正、负号数字化。通常的做法是用一位二进制表示符号, 称为“符号位”, 放在有效数字的前面, 用“0”表示正, 用“1”表示负。

带符号的二进制数在计算机中有三种表示形式: 原码、反码和补码。下面的讨论中假定字长为 8 位。

1. 原码、反码和补码的表示方法

(1) 原码 在数值的前面直接加一符号位的表示法称为原码表示法。例如数 +7 和 -7 的原码分别为:

	符号位	数值位
$[+7]_{\text{原}} =$	0	0000111
$[-7]_{\text{原}} =$	1	0000111

在这种表示法中, 数 0 的原码有两种形式, 即:

$$[+0]_{\text{原}} = 00000000 \quad [-0]_{\text{原}} = 10000000$$

若字长为 8 位, 则原码的表示范围为 $-127 \sim +127$; 若字长为 16 位, 则原码的表示范围为 $-32767 \sim +32767$ 。

(2) 反码 正数的反码与原码相同; 负数的反码, 符号位仍为“1”, 数值部分“按位取反”。如 +7 和 -7 的反码分别为:

$$\begin{aligned} [+7]_{\text{反}} &= 00000111\text{B} = 07\text{H} \\ [-7]_{\text{反}} &= 11111000\text{B} = \text{F8H} \end{aligned}$$

在这种表示法中, 数 0 的反码也有两种形式, 即:

$$[+0]_{\text{反}} = 00000000 = 00\text{H} \quad [-0]_{\text{反}} = 11111111 = \text{FFH}$$

字长为 8 位和 16 位时, 反码的表示范围分别为 $-127 \sim +127$ 和 $-32767 \sim +32767$ 。

(3) 补码

1) 模的概念: 把一个计量单位称之为模或模数, 用 M 表示。例如, 时钟是以 12 为计数循环, 即以 12 为模。在时钟上, 时针加上 (正拨) 12 的整数倍或减去 (反拨) 12 的整数倍, 时针的位置不变。14 点钟在舍去模 12 后, 成为 (下午) 2 点钟 ($14 = 14 - 12 = 2$)。从 0 点出发逆时针拨 10 格即减去 10h, 也可看成是从 0 点出发顺时针拨 2 格 (加上 2h), 即 2 点 ($0 - 10 = -10 = -10 + 12 = 2$)。因此, 在模 12 的前提下, -10 可映射为 $+2$ 。在讨论三角函数时以 360° 为计数循环, $410^\circ - 360^\circ = 50^\circ$ 。现实中还有许多以模为计数单位的例子。

计算机的运算部件与寄存器都有一定字长的限制, 因此它的运算也是一种模运算。例如一定位数的计数器, 在计满后会产生溢出, 又从头开始计数。产生溢出的量就是计数器的模。如前述 8 位二进制数, 它的模数为 $2^8 = 256$ 。

由此可以看出, 对于一个模数为 12 的循环计数系统来说, 加 2 和减 10 的效果是一样的。因此, 在以 12 为模的系统中, 凡是减 10 的运算都可以用加 2 来代替, 这就把减法问题化成加法问题了。10 和 2 对模 12 而言互为补数。在计算机中称为“补码”。

2) 补码的表示: 在补码表示法中, 正数的补码与原码相同; 负数的补码则是符号位为“1”, 数值部分按位取反后再在末位 (最低位) 加 1。

例如 $+7$ 和 -7 的补码分别为:

$$[+7]_{\text{补}} = 00000111\text{B} = 07\text{H}$$

$$[-7]_{\text{补}} = 11111001\text{B} = \text{F9H}$$

补码在微型机中是一种重要的编码形式, 请注意如下事项:

① 采用补码后, 可以方便地将减法运算转化为加法运算, 运算过程得到简化。因此, 计算机中带符号数一般采用补码表示。

② 正数的补码即是它所表示的数的真值, 而负数的补码的数值部分却不是它所表示的数的真值。

③ 采用补码进行运算, 所得结果仍为补码。为了得到结果的真值, 还得进行转换 (还原)。转换前应先判断符号位, 若符号为 0, 则所得结果为正数, 其值与真值相同; 若符号位为 1, 则应将它转换成原码, 然后得到它的真值。

④ 与原码、反码不同, 数值 0 的补码只有一个, 即 $[0]_{\text{补}} = 00000000\text{B} = 00\text{H}$ 。

⑤ 若字长为 8 位, 则补码所表示的范围为 $-128 \sim +127$; 若字长为 16 位, 则补码所表示的范围为 $-32768 \sim +32767$ 。

⑥ 进行补码运算时, 应注意所得结果不应超过上述补码所能表示数的范围, 否则会产生溢出而导致错误。采用其他码制运算时同样应注意这一问题。

2. 原码、反码和补码之间的转换 由于正数的原码、补码表示方法相同, 不存在转换问题。下面以负数情况分析。

(1) 已知原码, 求补码

例 13 已知某数 X 的原码为 10110100B , 试求 X 的补码。

解: 由 $[X]_{\text{原}} = 10110100\text{B}$ 知, X 为负数。求其补码表示时, 符号位不变, 数值部分按位求反, 再在末位加 1。