

21 世 纪 高 等 院 校 计 算 机 规 划 教 材

微机原理

与接口技术

主 编 周鸣争
主 审 孙家启



电子科技大学出版社

21 世纪高等院校计算机规划教材

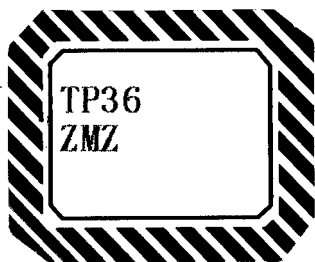
微机原理与接口技术

主 编 周鸣争

编 写 钱 峰 陈富春 周鸣争

阮进军 郑 刚

主 审 孙家启



电子科技大学出版社

内 容 简 介

本书从微型计算机系统的角度出发,较为全面地介绍了微型计算机的组成及各部分的工作原理。重点分析了 80X86 系统微处理器的基本结构、工作过程、基本指令系统以及汇编语言程序设计;阐述了计算机存储系统的组成与分类、内存和高速缓存的工作原理及典型芯片的应用、外部存储器的类型与工作原理以及存储器中的新技术;系统介绍了微型计算机系统总线结构和输入/输出技术,包括基本输入/输出方法、典型 I/O 接口芯片的应用等。同时覆盖了全国计算机等级考试以及全国高等学校计算机基础教育 PC 应用技术考试大纲的内容。

本书以理论与实践并重,传统与发展兼顾,教学与考试相容为原则;在强调基本概念的基础上,引入了大量的实例来阐明各种应用问题。本书可作为普通高等学校非计算机专业或高职高专计算机专业学生“微型计算机原理及应用”和“微机原理与接口技术”课程的教材,也可作为成人高等教育的培训教材及广大科技工作者的自学和考试参考书。

图书在版编目(CIP)数据

微机原理与接口技术/周鸣争主编. —成都:电子科技大学出版社,2005.8

ISBN 7-81094-902-0

I. 微... II. 周... III. ①微型计算机—理论—高等学校—教材 ②微型计算机—接口—高等学校—教材 IV. TP36

中国版本图书馆 CIP 数据核字(2005)第 094902 号

微机原理与接口技术

周鸣争 主编

出 版:电子科技大学出版社(成都建设北路二段四号 邮政编码:610054)

责任编辑:张 俊

发 行:电子科技大学出版社

印 刷:安徽省天歌印刷厂

开 本:787mm×1092mm 1/16 印张 21.875 字数 520 千

版 次:2005 年 8 月第 1 版

印 次:2005 年 8 月第 1 次印刷

书 号:ISBN 7-81094-902-0/TP·472

定 价:29.60 元

前 言

随着计算机技术和集成电路技术的发展,微型计算机在计算机应用中早已成为主流机型。“微型计算机原理及应用”是高等学校理工类各专业必修的一门重要技术基础课,它直接面向应用,在各学科中有着相当重要的作用。我们根据教育部全国高等学校计算机教学要求,同时兼顾全国高等学校(安徽考区)计算机基础教育教学(考试)新大纲的内容,结合我们近几年“微型计算机原理及应用”精品课程建设的教学实践,确定了本书的内容与体系结构。

本书的编写原则是:理论与实践并重,传统与发展兼顾,教学与考试相容。它具有这样一些特点:选材上力求广泛、实用、新颖;叙述上力求简洁、准确、易懂;鉴于非计算机专业学时少,本书编撰小而全、少而精,但全而不滥、少而不乏;以传统的16位微处理器Intel 8086/8088为主体展开全面讨论,同时又对发展过程中的现代PC机系统组成和微型计算机接口与应用作了详细介绍;书中特意安排的一些实例和各类题型的习题,既是成功经验的总结,可供实际应用参考,又是各种考试中频繁涉及的内容,对备考极具参考价值。

本书以高等学校非计算机专业本科和高职高专计算机专业学生为教学对象,系统介绍了微型计算机的组成原理、接口方法及应用技术。本书适合于全国高等学校计算机考试的需要,覆盖了四级微型机与接口技术考试的内容。

“微型计算机原理及应用”是一门实践性很强的课程,除课堂教学外,还应辅之以一定量的实验。作为教材,建议课内讲授50~60学时,实验20~30学时。

本书共十章,由周鸣争教授主编,由安徽省高等学校计算机教学指导委员会副主任兼秘书长孙家启教授主审。参加编写的人员有钱峰(第1章、第3章),陈富春(第2章、第8章),周鸣争(第4章、第6章),阮进军(第5章、第7章),郑刚(第9章、第10章)。全书由周鸣争最后统稿,强俊、童莉、楚宁等同志也参加了最后的统稿工作。

本书在编写过程中得到了有关专家热心的指导与无私的帮助,电子科技大学出版社为本书的尽快出版做了大量的工作,编者在此一并表示衷心的感谢。本书写作时参考了大量文献资料,在此也向这些文献资料的作者深表谢意。

由于时间仓促和水平所限,书中难免有不当和欠妥之处,敬请各位专家、读者不吝批评指正。

编 者
2005年6月

目 录

第 1 章 微型计算机基础知识	(1)
1.1 计算机发展过程与分类	(1)
1.1.1 计算机发展过程	(1)
1.1.2 计算机的分类和特点	(2)
1.1.3 微型机的分类与特点	(3)
1.2 微型机中的数制与码制	(3)
1.2.1 进位计数制及各计数制间的转换	(3)
1.2.2 二进制数的运算与带符号数的表示方法	(7)
1.2.3 机器内小数的表示方法	(13)
1.2.4 BCD 码(Binary Coded Decimal)	(14)
1.3 微型计算机系统	(15)
1.3.1 微型机系统硬件的基本组成	(15)
1.3.2 微型机系统的软件(Software)	(17)
1.3.3 微型机系统结构特点	(19)
1.3.4 微型机的性能指标	(20)
1.3.5 微型机的工作过程	(22)
习题	(23)
第 2 章 8086 微处理器	(25)
2.1 8086 CPU 的内部结构	(25)
2.1.1 执行部件 EU	(25)
2.1.2 总线接口部件 BIU	(26)
2.1.3 8086 CPU 的寄存器结构	(28)
2.2 8086 CPU 的外部引脚	(32)
2.3 8086 的工作模式与系统构成	(37)
2.3.1 8088/8086 CPU 的两种工作模式	(37)
2.3.2 两种模式下系统的典型配置	(37)
2.3.3 8086 对存储器的管理	(39)
2.4 8086 CPU 时序分析	(42)
2.4.1 为什么要研究时序	(42)
2.4.2 时序的概念	(42)
2.4.3 8086 的总线读/写操作时序	(44)
2.5 80X86 CPU 简介	(46)

习题	(53)
第 3 章 8086 的指令系统与程序设计	(55)
3.1 8086 的寻址方式	(55)
3.2 8086 的指令系统	(61)
3.2.1 概述	(61)
3.2.2 数据传输(Data Transfer)指令	(62)
3.2.3 算术运算(Arithmetic)指令	(68)
3.2.4 逻辑运算和移位、循环移位指令	(76)
3.2.5 控制转移指令	(79)
3.2.6 串操作指令	(86)
3.2.7 中断指令	(90)
3.2.8 处理器控制指令	(91)
3.3 汇编语言程序设计	(93)
3.3.1 汇编语言源程序的格式	(94)
3.3.2 汇编语言的语句	(96)
3.3.3 伪指令(指示性语句)	(101)
3.3.4 汇编程序设计	(106)
3.3.5 IBM PC-DOS 系统功能的调用	(125)
3.4 汇编语言程序的上机过程	(127)
3.4.1 建立、修改汇编语言源程序文件	(128)
3.4.2 汇编	(128)
3.4.3 链接	(129)
3.4.4 调试	(129)
习题	(132)
第 4 章 存储系统	(139)
4.1 概述	(139)
4.1.1 存储器的分类	(139)
4.1.2 存储器的主要性能指标	(140)
4.1.3 存储器的基本结构	(141)
4.1.4 存储系统的层次结构	(141)
4.2 半导体存储器	(142)
4.2.1 半导体存储器的分类	(143)
4.2.2 随机存取存储器 RAM	(144)
4.2.3 只读存储器 ROM	(148)
4.3 半导体存储器芯片与 CPU 的连接	(155)
4.3.1 存储器的扩展方法	(155)
4.3.2 CPU 与主存储器的连接	(156)
4.3.3 片选译码	(157)

4.3.4 存储器芯片同 CPU 连接时的关键问题	(161)
4.3.5 内存条技术	(164)
4.4 外存储器简介	(167)
4.4.1 软磁盘存储器系统	(167)
4.4.2 硬磁盘存储器系统	(168)
4.4.3 磁光盘	(169)
4.4.4 U 盘存储器	(171)
习题	(171)
第 5 章 基本输入输出接口	(173)
5.1 概述	(173)
5.1.1 接口基本概念	(173)
5.1.2 接口的主要功能	(173)
5.2 I/O 端口的编址方式	(174)
5.2.1 I/O 端口的概念	(174)
5.2.2 I/O 端口的编址方式	(175)
5.2.3 I/O 端口的地址译码	(176)
5.3 I/O 数据的传输方式	(178)
5.3.1 程序传送方式	(178)
5.3.2 中断传送方式	(181)
5.3.3 DMA 传送方式(直接存储器传送方式)	(182)
习题	(193)
第 6 章 中断与定时/计数器接口	(194)
6.1 概述	(194)
6.1.1 中断的基本概念	(194)
6.1.2 中断处理过程	(195)
6.1.3 中断优先级(中断优先权)	(198)
6.1.4 中断的嵌套	(200)
6.2 8086/8088 的中断系统	(201)
6.3 可编程中断控制器 8259A	(207)
6.3.1 8259A 功能与结构	(207)
6.3.2 中断的顺序	(210)
6.3.3 编程概述	(210)
6.4 可编程定时/计数器 8253	(220)
6.4.1 8253 功能和结构	(220)
6.4.2 8253 的初始化编程	(223)
6.4.3 8253 的工作方式	(224)
习题	(228)
第 7 章 串/并行通信及接口	(231)

7.1 概述	(231)
7.1.1 并行通信	(231)
7.1.2 串行通信	(232)
7.2 并行接口芯片 8255A	(235)
7.2.1 8255A 的内部结构和引脚信号	(235)
7.2.2 8255A 的控制字	(237)
7.2.3 8255A 的工作方式	(238)
7.2.4 8255A 的应用	(242)
7.3 可编程串行接口芯片	(245)
7.3.1 8251A 的内部结构	(245)
7.3.2 8251A 的控制字及初始化方法	(248)
7.3.3 8251A 的应用	(251)
习题	(253)
第 8 章 A/D 与 D/A 接口	(255)
8.1 概述	(255)
8.2 控制系统的模拟接口	(256)
8.3 D/A 和 A/D 芯片及其接口	(257)
8.3.1 D/A 转换器	(257)
8.3.2 A/D 转换器	(261)
习题	(268)
第 9 章 PC 机的总线技术	(270)
9.1 总线概述	(270)
9.1.1 总线和总线标准	(270)
9.1.2 总线的类型	(270)
9.1.3 总线仲裁控制	(271)
9.1.4 总线通信协议	(273)
9.2 常用系统总线	(274)
9.2.1 PC 总线	(274)
9.2.2 ISA 总线	(276)
9.2.3 EISA 总线	(277)
9.2.4 PCI 总线	(278)
9.2.5 AGP 总线	(281)
9.3 设备总线	(283)
9.3.1 IDE/ATA 总线	(283)
9.3.2 SATA 总线	(284)
9.3.3 SCSI 总线	(285)
9.3.4 USB 总线	(285)
9.3.5 IEEE-488 总线	(289)

习题	(291)
第 10 章 微型计算机系统	(292)
10.1 概述	(292)
10.2 系统控制逻辑及控制芯片组	(293)
10.2.1 什么是系统控制逻辑及控制芯片组	(293)
10.2.2 早期的芯片组	(293)
10.2.3 采用北桥/南桥体系结构的芯片组	(293)
10.2.4 采用 Hub 体系结构的芯片组	(295)
10.3 内存条	(297)
10.3.1 内存芯片 Bank 与芯片容量的新表示	(297)
10.3.2 内存条的组成	(298)
10.4 主板	(301)
10.5 微型计算机的体系结构	(303)
10.5.1 早期微型计算机的体系结构	(303)
10.5.2 控制芯片组出现初期的微型计算机的体系结构	(305)
10.5.3 采用北桥/南桥芯片组的微型计算机的体系结构	(305)
10.5.4 采用 Hub 芯片组的微型计算机的体系结构	(306)
10.6 3GIO 与下一代微型计算机体系结构简介	(310)
10.6.1 3GIO 的提出	(310)
10.6.2 3GIO 的主要特点	(311)
10.6.3 下一代微型计算机的体系结构	(312)
10.7 微型计算机中常用的接口卡	(313)
10.7.1 显示卡	(313)
10.7.2 声卡	(318)
10.7.3 调制解调器	(321)
10.7.4 网卡	(323)
习题	(324)
附录	(325)
附录 1 常用字符与 ASCII 代码对照表	(325)
附录 2 DOS 功能调用(INT 21H)	(326)
附录 3 中断向量地址一览	(331)
附录 4 Emu8086 简介	(333)
参考文献	(339)

第 1 章 微型计算机基础知识

电子计算机的产生和发展是 20 世纪人类最伟大的成就之一。现在,人们提到的计算机都是指电子数字计算机。由于计算机应用的广泛性,其高超的计算能力是人人皆知的。然而,从科学的角度来看,它仍然只是一种计算和信息加工的机器。由于它具有存储容量大、运算速度快和计算精度高等优点,使它远远胜过已有的其他计算工具。当然,作为计算和信息加工的工具,计算机与人类发明的其他计算工具相比,有了质的飞跃。它减轻并部分代替了人的脑力劳动,在特定时间内,能够完成人脑无法完成的工作量,也因此获得电脑的美称。

面对功能强大、结构复杂的现代计算机系统,要学习其组成原理,应从基础开始,循序渐进地学习,因此,本章将介绍一些必备的基础知识,具体包括:计算机中数据信息的表示、计算机的基本组成和工作原理以及微型计算机的基本结构。

1.1 计算机发展过程与分类

1.1.1 计算机发展过程

1946 年 2 月,从第一台电子计算机 ENIAC 在美国加州(由宾夕法尼亚大学研制)问世至今,已经历了四代发展过程。

第一代(1946~1959):电子管计算机时代,即内部元件使用的是电子管,主存储器采用磁鼓、磁芯等磁性材料,外存储器采用磁带,主要使用机器语言和汇编语言编制程序。这个时代的计算机主要用于科学和工程方面的计算。

第二代(1960~1964):晶体管计算机时代,即内部元件使用的是晶体管。主存储器仍然采用磁鼓、磁芯等磁性材料,外存储器开始使用磁盘,与第一代计算机相比软件有了很大的进步,出现了许多高级语言和编译程序。第二代计算机主要用于商业、企事业单位的事务管理、大学教学和工业控制等领域。

第三代(1965~1970):中小规模集成电路时代,即逻辑元件开始采用小规模和中规模集成电路(SSI 和 MSI),采用磁芯作为主存储器,计算机的性能有了较大的提高,应用领域日益扩大。随着集成电路规模的不断扩大出现了小型计算机,企事业单位的日常管理和工业控制逐渐进入小型机时代。在这个时期出现了多道程序设计和分时操作系统。

第四代(1971 至今):大规模和超大规模集成电路时代,使用半导体存储器;出现了用共享存储器、分布存储器或向量硬件选择的不同结构的并行计算机;诞生了微型计算机,使得计算机的体积越来越小,功能越来越强,性价比越来越好,应用范围越来越广;尤其是在办公自动化和家庭自动化等方面有了广泛的应用。

在有些资料上把第四代的后期划分出来,称它是第五代计算机(1991 至今):采用 VISL 工艺更加完善的高密度、高速度处理机和存储器芯片,它的最大特点是进行大规模并行处理,采用可扩展的和容许时延的系统结构。

1.1.2 计算机的分类和特点

电子计算机种类繁多,可以从不同的角度对计算机进行分类。

1.按计算机中信息的表示和处理方式的不同进行分类

电子计算机从发展开始就分为电子数字计算机和电子模拟计算机两大类。其区别在于计算机的表示形式和对信息的处理方式不同。

(1) 电子数字计算机是对离散量(数字量)进行运算的电子设备。在机器内部进行运算的是二进制数。电子数字计算机具有速度快、精确性高、存储信息量大等优点,适宜于科学计算、信息处理、过程控制、人工智能等方面,其应用范围非常广泛。人们通常所说的“计算机”,即是电子数字计算机的简称,常常也称之为“电脑”。

(2) 电子模拟计算机是对连续量进行运算的电子设备。被运算的对象有电压、电流、角度等连续变化的物理量。电子模拟计算机解题速度快,主要用于高阶微分方程求解及模拟控制系统等领域。其特点是电子模拟计算机中的信息不易存储,计算精度受设备精度的限制,所以其通用性不强,在日常工作中应用较少。

2.按计算机应用的范围进行划分

电子计算机按应用的范围大小可分为通用计算机和专用计算机两大类。

(1) 通用计算机是指能够装配各种应用软件,为了解决各种问题而设计的计算机。其功能强、配置全、用途广、通用性好,市场上销售的电子计算机基本上是通用计算机。

(2) 专用计算机是指为了解决一个或一类特定问题而设计的计算机。其功能单纯、使用面窄,主要应用在军事控制系统和工业生产过程控制中。

3.按计算机运算速度及软硬件配套规模划分

按计算机的字长、运算速度、存储容量、外部设备、输入输出能力、功能强弱及软硬件配套规模可以把计算机分为巨型机、大型机、中型机、小型机和微型机 5 类。根据美国电子工程师协会于 1989 年 11 月提出的标准,也可把计算机划分为巨型机、小巨型机、大型机、小型机、工作站和个人计算机等 6 类。

(1) 巨型机。巨型机也称为超级计算机,在所有计算机类型中价格最贵,功能最强,其浮点运算速度达到 1TFLOPS,即每秒万亿次的高性能计算机。它主要用于军事技术(如核武器等)和尖端科学(如空间技术,社会模拟,中、长期天气预报)等研究方面。巨型机的研究水平、生产能力及其应用程度,已成为衡量一个国家经济实力与科技水平的重要标志。

(2) 大型机。内存可达 1GB 以上,整机速度达 300MIPS~750MIPS,具有很强大的处理和管理能力,其使用范围比巨型机普遍,它是事务处理、商业处理、信息管理、大型数据库和数据通信的主要支柱。在计算机向网络迈进的时代,仍有着较广阔的生存空间。而巨型计算机已基本被小型机和大型机所代替,已失去了往日的“风采”。

(3) 小型机。其运算速度和规模比大型主机差,但功能上在向它们靠近。可在一般企业、事业、学校等单位使用。虽然,现代高档微型机和现代小型机的功能已没有多大差别,但有些高档小型机在速度、容量、外设和软件的完善性能上仍然占有一定的优势。目前微型机还不能完全代替小型机。小型机结构简单,可靠性高,成本较低,不需长期训练即可维护使用,这对于大、小用户来说比昂贵的大型主机具有更强大的吸引力。

(4) 工作站。它是介于PC机和小型机之间的一种高档微型机,是专长于处理某种特殊事务的计算机。它与网络系统中的“工作站”,在用词上相同,但含义不同。网络上的“工作站”常用于泛指联网用户的结点,以区别于网络服务器,这样的工作站只是一般的PC机。

(5) 微型机。这类计算机的功能及特点见下面的微型机分类与特点。

当然这种划分随着时间的推移也在变化,现在的奔腾系列微型机性能指标已达到早期的巨型机。目前的计算机朝着微型化和巨型化方向发展,前者价格每年在降低而性能在增强,后者代表了国家的科学技术水平。相对比较而言,大中型机的优势在渐渐消失,因而其应用范围也在不断地缩小。

1.1.3 微型机的分类与特点

可以从不同角度对微型机进行分类。通常按处理器位数和它的组装形式来分。

1. 按微处理器的位数来分

按微处理器的位数来分可分为4位、8位、16位、32位和64位微型机。所谓位数是指微处理器一次能够传送的最大数据位数。

2. 按微型机的组装形式来分

(1) 单片机:是指将微处理器、RAM、ROM、I/O接口电路、A/D、D/A和计时/计数器等部件集成在一块芯片中的计算机。

(2) 单板机:是指将微处理器、小容量的RAM、ROM和少量I/O接口集装在一块印刷电路板上,再装上适当的小键盘、发光二极管显示器,配上必要的监控程序而组成的微型机,就可完成一些简单的数据处理和编辑功能。如国内使用较多的TP-801(Z80)、TP-86(8086)、SDK-86(8086)等。

(3) 多板微型机:是指将微处理器、RAM、I/O接口电路、电源等组装在不同的印刷电路板上,然后装进同一个机箱,就构成了一个多板微型机。它可配置键盘、阴极射线管监视器(CRT)、打印机、软/硬磁盘驱动器等多种外部设备和足够的软件,而成为一个完整的微型机系统。

1.2 微型机中的数制与码制

1.2.1 进位计数制及各计数制间的转换

数制是人们对事物数量计数的一种统计规律。在日常生活中最常用的是十进制。但

在微型机中,由于电气元件最易实现的是两种稳定状态,即器件的“开”与“关”、电平的高与“低”,因此,采用二进制数的“0”和“1”能很方便地表示机内的数据运算与存储。在编程时,为了方便阅读、书写和记忆,人们还经常用八进制数或十六进制数来表示二进制数。1个数可以用不同计数制形式表示,但该数的大小则是相等的。

1. 进位计数制

进位计数制是采用位置表示法,即同一数字在不同的数位所代表的数值是不同的。每一种进位计数均包含着两个基本的因素:

①基数 R (Radix):它代表计数制中所用的数码个数。如:二进制计数中用到 0 和 1 两个数码;而八进制计数中用到 0~7 共 8 个数码。一般来说,基数为 R 的计数制(简称 R 进制)中,包含 0, 1, $\dots$, $R-1$ 个数码,进位规律为“逢 R 进一”。

②位权 W (Weight):进位计数制中,某个数位的值是由该位的数码值乘以处在这一位的固定常数决定的,我们把这一固定常数称之为位权值,简称位权。各位的位权是以 R 为底的幂。如十进制基数 $R=10$,则个位、十位、百位上的位权分别为 10^0 、 10^1 、 10^2 。

因此,一个 R 进制数 N ,可以用以下两种形式表示:

①并列表示法,或称位置计数法:

$$(N)_R = (K_{n-1}K_{n-2}\cdots K_1K_0K_{-1}K_{-2}\cdots K_{-m})_R$$

②多项式表示法,或称权展开式:

$$\begin{aligned}(N)_R &= K_{n-1}R^{n-1} + K_{n-2}R^{n-2} + \cdots + K_1R^1 + K_0R^0 + K_{-1}R^{-1} + \cdots + K_{-m}R^{-m} \\ &= \sum_{i=-m}^{n-1} K_i R^i\end{aligned}$$

其中, m, n 为正整数, n 代表整数部分的位数; m 代表小数部分的位数; K_i 代表 R 进制中的一个数码, $0 \leq K_i \leq R-1$ 。

(1) 二进制数

二进制数, $R=2$, K_i 可取 0 或 1, 进位规律为“逢二进一”。任一个二进制数 N 可表示为:

$$(N)_2 = K_{n-1}2^{n-1} + K_{n-2}2^{n-2} + \cdots + K_12^1 + K_02^0 + K_{-1}2^{-1} + \cdots + K_{-m}2^{-m}$$

例如: $(1001.101)_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$

(2) 八进制数

八进制数, $R=8$, K_i 可取 0~7 共 8 个数码中的任意一个, 进位规律为“逢八进一”。任意一个八进制数 N 可以表示为:

$$(N)_8 = K_{n-1}8^{n-1} + K_{n-2}8^{n-2} + \cdots + K_18^1 + K_08^0 + K_{-1}8^{-1} + \cdots + K_{-m}8^{-m}$$

例如: $(246.12)_8 = 2 \times 8^2 + 4 \times 8^1 + 6 \times 8^0 + 1 \times 8^{-1} + 2 \times 8^{-2}$

(3) 十六进制数

十六进制数, $R=16$, K_i 可取 0~15 共 16 个数码中的任一个, 但 10~15 分别用 A, B, C, D, E, F 表示, 进位规律为“逢十六进一”。任意一个十六进制数 N 可表示为:

$$(N)_{16} = K_{n-1}16^{n-1} + K_{n-2}16^{n-2} + \cdots + K_116^1 + K_016^0 + K_{-1}16^{-1} + \cdots + K_{-m}16^{-m}$$

例如: $(2D07.A)_{16} = 2 \times 16^3 + 13 \times 16^2 + 0 \times 16^1 + 7 \times 16^0 + 10 \times 16^{-1}$

注意:在实际表示中,一个十六进制数如果最高位数字是字母(A~F),则字母前必须

加一个数字 0,以便与变量名相区别。

以上 3 种进制数与十进制数的对应关系如表 1-1 所示。为避免混淆,除用(N)_R 的方法区分不同进制数外,还常用在数字后加字母作为标注。其中:字母 B(Binary)表示为二进制数;字母 Q(Octave 的缩写为字母 O,为区别数字 0 写为 Q)表示为八进制数;字母 D(Decimal)或不加字母表示为十进制数;字母 H(Hexadecimal)表示为十六进制数。

表 1-1 二、八、十、十六进制数码对应表

十进制	二进制	八进制	十六进制
0	0000B	0Q	0H
1	0001B	1Q	1H
2	0010B	2Q	2H
3	0011B	3Q	3H
4	0100B	4Q	4H
5	0101B	5Q	5H
6	0110B	6Q	6H
7	0111B	7Q	7H
8	1000B	10Q	8H
9	1001B	11Q	9H
10	1010B	12Q	0AH
11	1011B	13Q	0BH
12	1100B	14Q	0CH
13	1101B	15Q	0DH
14	1110B	16Q	0EH
15	1111B	17Q	0FH
16	10000B	20Q	10H

2. 各种进制数间的相互转换

(1) 各种进制数转换成十进制数

各种进制数转换成十进制数的方法是:将各进制数先按权展开成多项式,再利用十进制运算法则求和,即可得到该数对应的十进制数。

【例 1.1】 将数 1001.101B;246.12Q;2D07.AH 转换为十进制数。

$$1001.101B = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ = 8 + 1 + 0.5 + 0.125 = 9.625$$

$$246.12Q = 2 \times 8^2 + 4 \times 8^1 + 6 \times 8^0 + 1 \times 8^{-1} + 2 \times 8^{-2} \\ = 128 + 32 + 6 + 0.125 + 0.03125 = 166.15625$$

$$2D07.AH = 2 \times 16^3 + 13 \times 16^2 + 0 \times 16^1 + 7 \times 16^0 + 10 \times 16^{-1} \\ = 8129 + 3328 + 7 + 0.625 = 11527.625$$

(2) 十进制数转换为二、八、十六进制数

任一十进制数 N 转换为 q 进制数,须将整数部分与小数部分分别进行转换,然后再用小数点将已转换过的这两部分连接起来。

① 整数部分转换步骤:

第 1 步:用 q 去除 N 的整数部分,得到商和余数,记余数为 q 进制整数的最低位数码 K₀;

第2步:再用 q 去除上一步得到的商,求出新的商和余数,记余数为 q 进制整数的次低位数码 K_i ;

第3步:重复第2步,直至商为零,整数转换结束。此时,记余数为 q 进制整数的最高位数码 K_{n-1} 。

【例 1.2】将数 168 转换为二、八、十六进制数。

$$\begin{array}{r}
 2 \overline{) 168} \\
 \underline{2 84} \text{ 余数 } 0, K_0 = 0 \\
 \underline{2 42} \text{ 余数 } 0, K_1 = 0 \\
 \underline{2 21} \text{ 余数 } 0, K_2 = 0 \\
 \underline{2 10} \text{ 余数 } 1, K_3 = 1 \\
 \underline{2 5} \text{ 余数 } 0, K_4 = 0 \\
 \underline{2 2} \text{ 余数 } 1, K_5 = 1 \\
 \underline{2 1} \text{ 余数 } 0, K_6 = 0 \\
 \phantom{2 0} \text{ 余数 } 1, K_7 = 1 \\
 \therefore 168 = 10101000B
 \end{array}$$

$$\begin{array}{r}
 8 \overline{) 168} \\
 \underline{8 21} \text{ 余数 } 0, K_0 = 0 \\
 \underline{8 2} \text{ 余数 } 5, K_1 = 5 \\
 \phantom{8 0} \text{ 余数 } 2, K_2 = 2 \\
 \therefore 168 = 250Q \\
 \\
 16 \overline{) 168} \\
 \underline{16 10} \text{ 余数 } 8, K_0 = 8 \\
 \phantom{16 0} \text{ 余数 } 10, K_1 = A \\
 \therefore 168 = A8H
 \end{array}$$

②小数部分转换步骤:

第1步:用 q 去乘 N 的纯小数部分,得到乘积的整数部分,记为 q 进制小数的第1个数码 K_{-1} ;

第2步:再用 q 去乘上次乘积的纯小数部分,得到新乘积的整数部分,记为 q 进制小数的次位数数码 K_{-i} ;

第3步:重复第2步,直至乘积的小数部分为零,或者达到所需要的精度位数为止。此时,乘积的整数部分,记为 q 进制小数位的数码 K_{-m} 。

【例 1.3】将数 0.686 转换为二、八、十六进制数。

$$\begin{array}{l}
 0.686 \times 2 = 1.372 \quad K_{-1} = 1 \quad 0.686 \times 8 = 5.488 \quad K_{-1} = 5 \quad 0.686 \times 16 = 10.976 \quad K_{-1} = A \\
 0.372 \times 2 = 0.744 \quad K_{-2} = 0 \quad 0.488 \times 8 = 3.904 \quad K_{-2} = 3 \quad 0.976 \times 16 = 15.616 \quad K_{-2} = F \\
 0.744 \times 2 = 1.488 \quad K_{-3} = 1 \quad 0.904 \times 8 = 7.232 \quad K_{-3} = 7 \quad 0.616 \times 16 = 9.856 \quad K_{-3} = 9 \\
 0.488 \times 2 = 0.976 \quad K_{-4} = 0 \quad 0.232 \times 8 = 1.856 \quad K_{-4} = 1 \quad 0.856 \times 16 = 13.696 \quad K_{-4} = D \\
 0.976 \times 2 = 1.952 \quad K_{-5} = 1 \quad 0.856 \times 8 = 6.848 \quad K_{-5} = 6 \quad 0.696 \times 16 = 11.136 \quad K_{-5} = B \\
 \therefore 0.686 \approx 0.10101B \quad \therefore 0.686 \approx 0.53716Q \quad \therefore 0.686 \approx 1.AF9DBH
 \end{array}$$

【例 1.4】将数 168.686 转换为二、八、十六进制数。

依据例 1.2、例 1.3 可得:

$$168.686 = 10101000.10101B$$

$$168.686 = 250.53716Q$$

$$168.686 = A8.AF9DBH$$

由以上例子可以看出,二进制表示的数愈精确,所需的数位就愈多,很不方便书写和记忆,且易出错。若用同样数位表示数,八、十六进制数所表示数的精度高。所以在汇编语言中常用八进制或十六进制数作为二进制数的编码,这样书写方便,且易于记忆。

(3)二进制数与八进制数之间的相互转换

因为 $2^3=8$, 所以采用“合三为一”的原则, 从小数点开始分别向左、右两边各以 3 位为一组进行二→八换算, 若不足 3 位以 0 补足, 即可将二进制数转换为八进制数。

【例 1.5】 将数 1111011.0101B 转换为八进制数。

$$\begin{array}{rcccccc} \because & 001 & 111 & 011 & . & 010 & 100 \\ & 1 & 7 & 3 & . & 2 & 4 \\ \therefore & 1111011.0101\text{B} & = & 173.24\text{Q} \end{array}$$

反之, 采用“一分为三”的原则, 每位八进制数用 3 位二进制数表示, 即可将八进制数转换为二进制数。

【例 1.6】 将数 1357.246Q 转换成二进制数。

$$\begin{array}{rccccccccc} \because & 1 & 3 & 5 & 7 & . & 2 & 4 & 6 \\ & 001 & 011 & 101 & 111 & . & 010 & 100 & 110 \\ \therefore & 1357.246\text{Q} & = & 1011101111.01010011\text{B} \end{array}$$

(4) 二进制数与十六进制数之间的相互转换

因为 $2^4=16$, 所以采用“合四为一”的原则, 从小数点开始分别向左、右两边各以 4 位为一组进行二→十六换算, 若不足 4 位以 0 补足, 即可将二进制数转换为十六进制数。

【例 1.7】 将数 1101000101011.001111B 转换为十六进制数。

$$\begin{array}{rccccccccc} \because & 0001 & 1010 & 0010 & 1011 & . & 0011 & 1100 \\ & 1 & A & 2 & B & . & 3 & C \\ \therefore & 1101000101011.001111\text{B} & = & 1\text{A}2\text{B}.3\text{CH} \end{array}$$

反之, 采用“一分为四”的原则, 每位十六进制数用 4 位二进制数表示, 即可将十六进制数转换为二进制数。

【例 1.8】 将数 4D5E.6FH 转换成二进制数。

$$\begin{array}{rccccccccc} \because & 4 & D & 5 & E & . & 6 & F \\ & 0100 & 1101 & 0101 & 1110 & . & 0110 & 1111 \\ \therefore & 4\text{D}5\text{E}.6\text{FH} & = & 100110101011110.01101111\text{B} \end{array}$$

1.2.2 二进制数的运算与带符号数的表示方法

1. 二进制数的算术运算

二进制数不仅物理上容易实现算术运算, 而且运算规则也比较简单, 其加、减法遵循“逢二进一”、“借一当二”的原则, 和、差、积的运算规律如下:

$$\begin{array}{lll} 0+0=0 & 0-0=0 & 0\times 0=0 \\ 0+1=1 & 0-1=1 \text{ 有借位} & 0\times 1=0 \\ 1+0=1 & 1-0=1 & 1\times 0=0 \\ 1+1=10 \text{ 有进位} & 1-1=0 & 1\times 1=1 \end{array}$$

下面将通过 4 个例子来说明二进制数的加、减、乘、除运算过程。

【例 1.9】 求 $11001010\text{B} + 11101\text{B}$ 。

$$\begin{array}{r}
 \text{被加数} \quad 11001010 \\
 \text{加数} \quad +) \quad 11101 \\
 \hline
 \text{和} \quad 11100111
 \end{array}$$

$$\therefore 11001010\text{B} + 11101\text{B} = 11100111\text{B}$$

由此可见,两个二进制数相加时,每一位有 3 个数参与运算(本位被加数、加数、低位进位),从而得到本位的和,以及向高位的进位。

【例 1.10】求 $10101010\text{B} - 10101\text{B}$ 。

$$\begin{array}{r}
 \text{被减数} \quad 10101010 \\
 \text{减数} \quad -) \quad 10101 \\
 \hline
 \text{差} \quad 10010101
 \end{array}$$

$$\therefore 10101010\text{B} - 10101\text{B} = 10010101\text{B}$$

由此可知,二进制减法与加法类似,每一位有 3 个数参与运算(本位被减数、减数、低位借位),从而得到本位的差,以及向高位的借位。

【例 1.11】求 $110011\text{B} \times 1011\text{B}$ 。

$$\begin{array}{r}
 \text{被乘数} \quad 110011 \\
 \text{乘数} \quad \times) \quad 1011 \\
 \hline
 110011 \\
 110011 \\
 000000 \\
 +) \quad 110011 \\
 \hline
 \text{积} \quad 1000110001
 \end{array}$$

$$\therefore 110011\text{B} \times 1011\text{B} = 1000110001\text{B}$$

由此可知,二进制数乘法与十进制数乘法相类似,可用乘数的每一位去乘被乘数,乘得的中间结果的最低有效位与相应的乘数位对齐,若乘数位为 1,则中间结果为被乘数;若乘数位为 0,则中间结果为 0,最后把所有中间结果同时相加即可得到乘积。这种算法对微型机实现很不方便。在没有乘法指令的微型机中,常采用被乘数左移或部分积右移的方法编程,来实现乘法运算。目前 8086 以上微型机均有专门的乘法指令来完成乘法(详见本书第三章指令系统的介绍),给用户带来许多方便,提高了机器的运算速度。

【例 1.12】求 $100100\text{B} \div 101\text{B}$ 。

$$\begin{array}{r}
 1110 \\
 101 \overline{) 100100} \\
 \underline{101} \\
 1000 \\
 \underline{101} \\
 110 \\
 \underline{101} \\
 11
 \end{array}$$

$$\therefore 100100\text{B} \div 101\text{B} = 1110\text{B} \text{ 余 } 11$$