

微型计算机原理及应用

薛钧义 虞鹤松 张彦斌



西安交通大学出版社

微型计算机原理及应用

薛钧义 虞鹤松 张彦斌

西安交通大学出版社

内容简介

本书是为电类非计算机专业本科生编写的属于第二层次“计算机硬件技术基础”的教材。主要内容有:计算机基础知识,计算机系统的基本结构与组成原理,结合 80X86 微处理器介绍指令系统、寻址方式、汇编语言及汇编程序设计;存储器及体系结构;常用的接口芯片及其与 CPU 的连接方法、接口技术。简要介绍了高档微机的多功能外围芯片;微型计算机系统组成及操作系统。教材内容选取上注意基础性、稳定性又不失先进性。编写中注意由浅入深,与实际应用相结合。每章都有习题及思考题。

本书适合于高等院校电类非计算机各专业作教材其学时数为 56~72 学时。也可供科技人员作为自学教材或参考书。

图书在版编目(CIP)数据

微型计算机原理及应用/薛钧义等编著. -西安:西安交通大学出版社,2000.7
ISBN 7-5605-1158-9

I. 微… II. 薛… III. 微型计算机 IV. TP36

中国版本图书馆 CIP 数据核字(2000)第 64805 号

■

*

西安交通大学出版社出版发行
(西安市咸宁西路 28 号 邮政编码:710049 电话:(029)2668316)
陕西宝石兰印务有限责任公司印装
各地新华书店经销

*

开本:787 mm×1 092 mm 1/16 印张:27.25 字数:671 千字
2000 年 7 月第 1 版 2000 年 7 月第 1 次印刷
印数:0001~5 000 定价:36.00 元

若发现本社图书有倒页、白页、少页及影响阅读的质量问题,请去当地销售部门调换或与我社发行科联系调换。发行科电话:(029)2668357,2667874

前 言

本书是为适应计算机技术的迅速发展,参照国家教委关于《加强工科非计算机专业计算机基础教学工作的几点意见》和工科计算机课程教学指导委员会提出的“文化基础、技术基础、应用基础”三层次教学体系的构想,组织编写的属于第二层次“计算机硬件技术基础”的本科生教材。

“微型计算机原理及应用”是电类非计算机专业学生掌握计算机硬件和软件(汇编语言)的入门课程。为此,本书在教材内容的选取与组织上掌握以下几点编写原则:

(1) 注意基础及实用性。重点介绍一般计算机系统的基本结构与组成原理。基础内容不针对某种机型,这样能使教学基本内容有相对的稳定性;另外,还注意系统性和先进性、追踪微机及应用技术的新水平与趋向、使教材内容又不失先进性。

(2) 淡化 CPU 内部的结构组成。从编程应用角度介绍 CPU 的功能、系统的连接、接口技术与应用。为微机的应用以及硬件的扩展、开发等后续的系列课程打下一定的基础。

(3) 根据目前国内高校的教学与实验条件、应用的主流机型,选择 80X86 微处理器来具体讲述 CPU 的体系结构、寻址方式、指令系统、存储器及其与 CPU 的连接、汇编语言程序设计、接口部件及接口技术和微机系统。

(4) 接口技术是组成任何微机应用系统的关键技术。目前 80386/486 微机系统都有相应配套的多功能外国芯片,如 82380 系列。83C490 系列。用这种多功能芯片要讲述清楚接口技术是较为困难的。我们认为仍应以常用的可编程接口电路芯片,如 8255、8253、8250、8237 为典型的介绍对象,以便于学生能较好掌握软、硬件相结合的可编程接口技术。而多功能外围芯片是上述芯片功能的集成,在此基础上作功能简介是容易理解的。

(5) 电类专业学科较多,对本课程的要求各异、有不同的教学时数。本教材在内容取材上尽量做到教学内容模块化,有些章节内容可以选学。

最后应指出,“微型计算机原理及应用”是实践性很强的课程,应重视实践性教学和上机能力的培养。使部分教学内容在上机实践中获取,这更有利于调动学生学习的主动性与积极性。本课程的建议学时为 56~72 学时范围。

本书共 9 章,由薛钧义同志主编并负责编写第 2、4、7 章;张彦斌同志编写第 1、3 章;虞鹤松同志编写第 5、6、8、9 章。宋卫平、王鹏光、邓雷、刘晓娟、常弘、宁改娣老师也参加了部分编写、实验工作。全书由薛钧义同志统编。

需要说明的是,考虑到读者的使用习惯和现状,本书在涉及软、硬盘及驱动器的尺寸时,仍使用了英寸表示,按国家标准换算后为:5.25 英寸=133.35mm,3.5 英寸=88.9mm。

本书在编写中参阅不少参考书及资料,学习和吸取了不少经验。出版过程中,得到了校、院领导的具体帮助与指导,得到了交大出版社的大力支持,并提出了许多宝贵意见,在此一并表示衷心感谢。

由于编者水平所限,加之时间仓促、错误与不当之处难免、恳请批评指正。

编著者

目 录

第 1 章 微型计算机概述	(1)
第 1 节 微型计算机的发展	(1)
第 2 节 微型计算机的应用	(3)
第 3 节 计算机中信息的表示与运算	(5)
第 4 节 80X86 处理器数据类型	(22)
第 2 章 微型计算机的基本组成及工作原理	(26)
第 1 节 指令格式	(26)
第 2 节 指令的寻址方式	(29)
第 3 节 指令的类型及功能	(38)
第 4 节 中央处理器的功能及组成	(44)
第 5 节 简单的程序编制与工作过程	(47)
第 6 节 组合逻辑控制器简介	(56)
第 7 节 可编程逻辑阵列 PLA	(57)
第 8 节 微程序控制器简介	(59)
第 9 节 计算机系统结构的新发展	(64)
第 3 章 半导体存储器	(69)
第 1 节 概述	(69)
第 2 节 典型的 RAM、ROM 芯片介绍	(72)
第 3 节 存储器与 CPU 的连接	(83)
第 4 节 存储器的体系结构	(95)
第 5 节 高速缓冲存储器	(99)
第 6 节 虚拟存储器	(101)
第 7 节 存储器技术的新发展	(102)
习题与思考题	(104)
第 4 章 Intel 80X86 系列微处理器	(106)
第 1 节 8086/8088 微处理器	(106)
第 2 节 80186 和 80286 微处理器	(135)
第 3 节 80386 微处理器	(136)
第 4 节 80486 微处理器	(147)
第 5 节 80386/80486 的存储器管理	(150)
第 6 节 80X86 的寻址方式	(158)
第 7 节 80X86 指令系统	(164)

习题与思考题	(200)
第5章 汇编语言程序设计	(204)
第1节 汇编语言语句格式	(204)
第2节 汇编语言程序的基本结构与设计方法	(213)
第3节 宏汇编	(237)
第4节 BIOS 中断和 DOS 系统功能调用	(240)
第5节 汇编语言与高级语言混合编程	(244)
习题与思考题	(248)
第6章 输入/输出方式及其中断处理	(251)
第1节 输入/输出方式概论	(251)
第2节 输入/输出接口基本结构及编址方式	(256)
第3节 80X86 的中断系统	(261)
第4节 中断控制器 8259A	(268)
习题与思考题	(276)
第7章 常用输入/输出接口电路	(277)
第1节 8255A 并行接口电路	(277)
第2节 可编程计数/定时器 8253	(291)
第3节 DMA 控制器 8237A-5	(307)
第4节 串行通信及串行接口 INS8250	(326)
第5节 多功能外围接口电路 82380、80350	(356)
习题与思考题	(360)
第8章 微型计算机系统	(362)
第1节 80X86 系列微机主机系统	(362)
第2节 微机系统常用外围设备接口技术	(368)
第3节 计算机网络概述	(376)
习题与思考题	(384)
第9章 微型计算机操作系统	(385)
第1节 计算机操作系统概论	(385)
第2节 微机操作系统 MS-DOS	(387)
第3节 微机操作系统 Windows	(397)
习题与思考题	(404)
附录1 80X86 的指令系统表	(405)
附录2 常用总线接口电路型号、名称	(429)
主要参考文献	(430)

第1章 微型计算机概述

电子计算机的发明是 20 世纪人类最辉煌的成就之一。自从 1946 年第一台计算机问世, 仅仅 50 多年, 电子计算机已深入到国民经济、科学技术、社会生产和人类生活的各个方面, 产生了极其深远的影响和难以估量的推动作用, 从而使人类进入了一个新的时代——信息革命的时代。

短短 50 多年, 电子计算机已经历了电子管计算机、晶体管计算机、集成电路计算机、大规模集成电路计算机整整四代的发展和变迁。作为开发工具的程序设计语言也由机器语言、汇编语言发展到高级语言以及在高级语言基础上集成的模块化语言。现在, 随着集成电路规模的迅速扩大, 计算机也趋于向两端发展, 即研制运算速度超过亿次的巨型计算机和以微处理器为核心组装的微型计算机。本书主要讨论微型计算机原理及其相应的接口技术。

第1节 微型计算机的发展

在计算机科学的发展过程中, 微型计算机的出现是划时代的进步, 它为计算机的发展和普及开辟了一条崭新的途径。由于微型计算机性能/价格比在各种机型中占有领先地位, 并以物美价廉、小巧灵活而深受用户欢迎, 所以它的发展速度大大超过了前几代计算机。自从美国 Intel 公司 1971 年研制出世界上第一台微处理器 Intel 4004 以来, 微型计算机已经历了五个发展时代, 如表 1.1 所示。第一代是微处理器萌芽阶段, 以 Intel 4004 和 8008 为代表; 第二代是设计和生产技术已进入成熟的阶段, 前期生产的 8 位中档机, 以 Intel 8080 和 MC6800 为代表, 后期生产的 8 位高档机, 是一种向 16 位机过渡的产品, 以 Z80, Intel 8085 和 MC6809 为代表; 第三代是 16 位微型计算机的发展阶段, 有代表性的 3 种芯片是 Intel 8086/8088、MC6800 和 Z8000, 其性能已可与过去中档小型计算机相比; 第四代是 32 位微型计算机的大发展时期, 特别是 80 年代后期推出的 Intel 80486, MC68030 和 MC68040 等, 均是高档 32 位微处理器。这些高档 32 位微处理器的显著特点是吸取了大中型计算机的体系结构的优点和采用了先进的超大规模集成电路 SLSI(Super Large Scale Integration)技术以及多级流水线和虚拟存储管理技术, 从而使其集成度和运算速度更上一层楼。IBM-PC/80486 和 IBM-PC/80586 就是采用这种高性能 32 位微处理器做成的典型的个人计算机。从 1992 年以来, 微处理器进入了第五个发展阶段, 即 64 位微处理器发展时代。这些芯片的最主要特点是采用了整数嵌入技术和浮点运算器以及超通道技术, 每个机器周期可执行 2 条指令, 因而芯片的集成度和运算速度均大大优于 32 位微处理器。如 Intel 公司的 Pentium-100 集成度高达 510 万只晶体管/片。此外, 由于整数和浮点运算部件采用了超级流水线式结构, 使得微处理器的运算速度展现了超级计算的新面貌, 性能达到了现有巨型机的水平。更值得一提的是, 在激烈的市场竞争中, 微处理器本身的体系结构也发生了重大变化。20 世纪 80 年代发展起来的针对传统的复杂指令

集计算机(Complex Instruction Set Computer – CISC)所存在的弊病,在体系结构上作了重大革新的精简指令集计算机(Reduced Instruction Set Computer – RISC),显示出了强大的生命力。IBM 公司推出的 Power PC 则是彻底的 RISC 结构,而于 1994 年 10 月发表的 Power PC620 全 64 位微处理器集成度已近 700 万只晶体管/片,由于局部采用的“超标量推测执行(MES)”的新一代高速体系结构,大大提高了速度。RISC 技术上的突破性进展,代表着微处理器的一个重要发展方向。可以相信,随着特大规模集成电路 ULSI(Ultra Large Scale Integration)和巨大规模集成电路 GLSI(Great Large Scale Intergration)的飞速发展,64 位微处理器技术还会向纵深发展。而以上述高性能微处理器为基础的超级微机的研制和开发也正是方兴未艾,前途无量。

表 1.1 微处理器和微型计算机的分代

特 点 项 目	时代 第一代 1971~1973 年	第二代 1974~1978 年	第三代 1978~1981 年	第四代 1981~1992 年	第五代 1992 年~
制造工艺 集成技术	PMOS LSI	NMOS LSI	HMOS/NMOS LSI/VLSI	NMOS/CMOS SLSI	NMOS/CMOS SLSI/ULSI
集成度	0.12~0.2 万 晶体管/片	0.5~0.9 万 晶体管/片	2~6.8 万 晶体管/片	10 万以上 晶体管/片	80 万以上 晶体管/片
字长(位)	4/8	8	16	32	64
基本指令 执行时间	10~20 μ s	1~1.3 μ s	<1 μ s	<125ns	<10ns
引脚数(条)	16/24	40	40~68	64~100	
典型微处理器	Intel 4004 Intel 8008 TMS 1000 PPS-4	Intel 8080 Intel 8085 M6800 M6809 Z80	Intel 8086/ 80386 Z8000 M68000 LSI-11/23 Intel 80186/ 80286	Intel 80386 Intel 80486 Z8000 HP9000 M68020	Alpha 21064/ 21164 R4000/8000 PA-RISC7100 Pentium-100 Power PC601
典型微型 计算机	MCS-4 MCS-8	CS 系列 H89 TRS-80 APPLE-II	IBM-PC/XT IBM-PC/AT HP98360CT Intel 86/330	IBM-PC/486 IBM-PC/586 LTE Elitexz 400E Win DX4/75	DEC 300/500 Axp IBM RS6000 500 HP9000/765

注:LSI(Large Scale Integration)大规模集成电路;VLSI(Very Large Scale Integration)甚大规模集成电路;
SLSI(Super Large Scale Integration)超大规模集成电路;ULSI(Ultra Large Scale Integration)特大规模
集成电路。

由于微处理器是微型计算机的核心构件,它在很大程度上决定了微型计算机及其系统的性能指标,所以,微型计算机的发展历程,从根本上看也就是微处理器的发展历程。现在,以4位,8位,16位微处理器芯片构成的所谓低档微型计算机,已广泛用在家用电器、仪器仪表和过程控制等领域,成为它们不可缺少的组成部分。在这方面特别值得一提的是被称作“微控制器(Micro Controller)”和“嵌入式计算机(Embedded Computer)”的8位和16位单片微型计算机,以其体积小、功耗低、功能强、价格低廉和可靠性高而倍受用户青睐,并且不断有一些通用和专用的新产品投放市场,以满足各方面和各个领域的需要。所以,这种低档微型计算机在未来计算机发展中一定会经久不衰,具有旺盛的生命力。32位微型计算机还正处在进一步完善和发展阶段,新的产品不断涌现,尤其是软件的发展。除做成通用的微型计算机系统如IBM-PC/486,IBM-PC/586外,一些专用32位微型计算机常作成工作站和网卡,活跃在信息管理系统和通信系统中,如SUN公司的SPARC station LX,IBM公司的RS6000M20等。而专门用于信息处理的数字信号处理器(DSP),也有了突飞猛进的发展,如TI公司的TMS320C80DSP,它将4个32位DSP,一个32位RISC主处理器、1个传输控制器、2个视频控制器和50KB SRAM集成在一个芯片上,构成了32位的专用微控制器。这种专用的DSP产品,可广泛用于通信、语音、图像等信号处理领域中。目前,64位微处理器制成的工作站如HP9000-765、IBM RS6000-570等也已投入市场。由于64位微型机具有比32位机更高的运算速度、更大的主存容量和更强的图形功能,因此尽管其软件还在发展之中,但却必将成为21世纪的微型计算机发展的主流。

与此同时,近年来多微处理器系统的发展也有了长足的进步,是微型计算机发展的另一动向。这是一种有多个微处理器并行运算的系统,其运算速度和工作性能不仅取决于所用微处理器的类型,而且和所用微处理器的数量有关。如Intel公司采用了30个Intel 80386研制成的IPSC机,其性能相当于IBM3090系列中最高档的大型机,价格只有后者的1/10。面对这一咄咄逼人的形势,迫使许多大型机和巨型机生产厂家也采用了与多处理器系统类似的结构——并行结构。如目前已投入运行的超级计算机Ncube2就包含了8192个微处理器,现在由65000个微处理器组成的Ncube3型机也在研制中。

展望未来,计算机的发展必将有很多新的突破。光学技术、超导技术、仿生技术的相互结合,必然产生一种全新的计算机,而人工智能的研究正在促进计算机面临一场新的革命。人工智能计算机,人工神经网络计算机……,这些“新一代”计算机的争先问世,都为21世纪人类大规模进入智能信息社会作了必要的准备。现在,一个国家计算机的发展水平及其应用的深度和广度,已经成为这个国家现代化水平的重要标志之一。毫无疑问,计算机在未来社会中将会发挥越来越重要的作用。

第2节 微型计算机的应用

微型计算机已无孔不入地渗透到所有应用领域,如表1.2所示。按照计算机加工信息的方式和处理信息的特点可分为两大类:数值应用和非数值应用,现在非数值应用已远远超过了数值计算。归纳起来,主要有以下方面:

1. 科学计算

计算机最早用于军事和科学计算。第一台计算机就是为解决复杂的弹道计算而研制的,

此后如人造卫星、导弹和航天飞机、原子反应堆、天气预报、大型桥梁、高层建筑、地震预报、飞机和轮船的外形设计等都离不开大型高速计算机。今天,随着微型机技术的飞速发展,特别是并行多处理机系统的诞生,使微型机性能已达到了大、中型计算机水平,而造价却比它们低得多,因此,采用微型计算机进行科学计算已是最理想的选择。

表 1.2 微型计算机的分代及各代的应用领域

特 点 项 目	时代 第一代 1971~1973 年	第二代 1974~1978 年	第三代 1978~1981 年	第四代 1981~1992 年	第五代 1992 年~
应用领域	1. 家用电器 2. 计算器 3. 简单控制	1. 智能终端 2. 仪器仪表 3. 工业控制 4. 教学 5. 数据处理	1. 实时控制 2. 数据库 3. 事务处理 4. 科学计算 5. 分布式系统 6. 局域网	1. 事务处理 2. 多用户数 据处理 3. 科学计算 4. 多微处理 器系统 5. 局域网 6. 工作站	1. 图形工作站 2. 大型计算 机网中的 服务器 3. 信息管理 系统 4. 大、中型机 的构件

2. 实时控制和仪器仪表智能化

微型计算机在工业生产过程控制、数控机床、仪器仪表控制、家用电器和民用产品控制方面已得到十分广泛的应用,并且获得了非常好的效果。

近年来在机械、冶金、石油、化工、电力、建筑及轻工业等各个部门,以微机为核心的计算机监测和控制系统随处可见。而专门用于工业测控领域的计算机——工业控制计算机也已形成系列产品,比如 STD 总线和 PC 总线工业控制微型计算机,基于 PC/104 总线的一种嵌入式工业控制机等,已在我国广泛使用。而以微型机技术为基础的可编程序控制器 PC 或 PLC,已经成为工业自动化领域广泛应用的自动化装置。所有这些对普及计算机的工业应用都起了很大的推动作用。

此外,基于微型计算机的机电仪一体化技术和产品已经成为热门话题。微型计算机的应用推动了仪器仪表的智能化,导致了一批新仪器的诞生。而由微处理器控制的洗衣机、冰箱、自动报时报警器等已经进入普通家庭。这方面,作为“微控制器”的单片计算机的应用更是十分活跃。显然,微型机在对促进传统产业的技术改造,加速新型产业的新兴发展中,必然产生越来越重要的影响。

3. 数据处理和信息管理

数据处理通常是指计算机对实时采集和人工送入的大量数据进行加工处理、转换分析、显示打印、反馈控制和发送远传的过程,这在航空、航天、邮电通信、军事科学和工业控制中应用非常广泛。如地面卫星接收系统,防空警戒雷达系统等。数据处理是一切信息管理、辅助决策的基础,信息管理即是在此基础上才对信息进行分类检索、查找统计、绘制图表和输出打印的。信息管理系统可由单个高档微机系统组成,也可以是不同类型的计算机网络系统,如飞机订票系统、情报检索系统、办公自动化、电子邮件系统等。据统计,用于数据处理的计算机在所有应用方面是占比例最大的,而微型计算机在该领域更是大有可为。

4. 在 CAD/CAM 中的应用

计算机在辅助设计和辅助制造(CAD/CAM)中的应用是一个十分重要和广泛的领域。利用计算机绘图不仅速度快,质量高、修改方便,还可以在任意方位,不同颜色下观察修改将要制造的产品结构、外形,大大提高设计的效率。设计好的产品可以利用计算机自动进行加工,这一过程通常由数控机床或数控加工中心来完成,数控机床是 CAM 的中心设备。通常把 CAD/CAM 放在一起。形成 CAD/CAM 一体化系统。目前我国 CAD 使用较为普遍,尤其在建筑、造船、机械制造和飞机制造行业中使用更为广泛。

5. 多媒体系统和信息高速公路

多媒体系统是一种除了能处理文字和数字外,还能处理图形、影像、动画、声音和视频信号等多种媒体的微机系统,并能和外部世界进行多功能和多用途的信息交流。目前,多媒体技术正朝着实用化、标准化方向发展,它的应用领域正在不断拓宽。除了能把各种家用电器的功能融为一体外,还可以用于商场购物指南、旅游导向、新产品演示、印刷出版、检测、教育培训、医疗诊断、科学研究等。在通信领域中,一旦实现了多媒体技术与通信卫星的合理搭配,人们就可以随时地获取世界各地的信息。学术界认为,多媒体技术是比当年电话技术、电视技术的出现意义更为重大的一次信息革命。现在,随着多媒体电脑、多媒体电视的出现,多媒体产品正在步入寻常百姓家。

当今社会信息量增长很快,特别是多媒体技术出现后,更使信息的流量猛增,而“信息堵塞”的现象也越来越突出,如何解决信息传输中“车多路窄”的状况,人们终于找到了“信息高速公路”这个法宝。“信息高速公路”是在全国范围内,以光缆为信息传输的主干线,同时采用支线光纤和多媒体终端,用“交互”方式传输数据、电视、话音、图像等多种形式信息的千兆(10^9)比特的高速数据网。是一种集计算机技术、无线电技术、声音处理技术、文件传输和可视通信技术于一体的高新技术系统。美国、日本、西欧以及发展中国家和地区都在争先恐后地计划、组织和实施“信息高速公路”的建设。其中美国于 1982 年所建设的交互网络 INTERNET 网,是目前国际上最大的计算机网络系统,被认为是正在蓬勃兴起的“信息高速公路”的支柱。我国已与 INTERNET 网实现了互连,并在全国重点高等院校和一些科研单位建立和开通了自己的“教育与科学研究网络”。这些虽然还只是“信息高速公路”的雏形,但其发展前景十分诱人。“信息高速公路”的建成,将使人们通过电信网坐在家里的电脑终端旁进行高效工作,以节省办公用房、缓解交通拥挤和减少城市污染;它可以给人们提供先进的社会服务和消灭城市和乡村之间实际存在的教育质量的差别;它还可以帮助人们把城市的高质量医疗服务送到偏远的山区等等。随着“信息高速公路”的建设,将进一步刺激多媒体市场的飞跃发展,并将兴起一个崭新的行业——信息高速公路业,从而产生巨大的社会效益,给人类社会带来巨大的冲击与变革。

除以上应用范围外,计算机还在辅助教学(CAI)、系统仿真与智能模拟、模式识别及人工智能等方面有着广泛的应用。

第3节 计算机中信息的表示与运算

计算机的最重要的功能是处理信息,如数值、文字、符号、语言、图形和图像等。在计算机内部,各种信息都必须采用数字化编码的形式被传送、存储和加工。其中把能够进行算术运算

得到明确数值概念的信息称为计算机数值数据,其余的信息称为非数值数据,本节介绍计算机中的数值数据和非数值数据信息的表示方法。

一、无符号数的表示及数制转换

(一)、无符号数的表示法

无符号数以不同的进位计数制来表示。所谓进位计数制是指用数字符号排列成数位,按由低位到高位进位方式计数来表示数的方法。进位计数制有多种,在计算机中经常使用的有十进制、二进制、八进制和十六进制。

进位计数制(简称计数制或进位制)最重要的两个基本要素是基与权。计数制中所使用的数码的个数称为基;计数制每一位所具有的值称为权。

1. 十进制数

基数为 10,需要 10 个数字符号 0,1,2,⋯,9。

各位的权是以 10 为底的幂,第 i 位的权为 10^i 。

所遵循的计数规律为:逢十进一。

对于任一个十进制数 N ,设整数部分有 n 位,小数部分有 m 位,则可表示为

$$(N)_{10} = \sum_{i=n-1}^{-m} D_i \times 10^i \quad (D_i \text{ 是 } 0,1,2,\dots,9 \text{ 中的一个}) \quad (1.3.1)$$

如: $(128.6)_{10} = 1 \times 10^2 + 2 \times 10^1 + 8 \times 10^0 + 6 \times 10^{-1}$

式(1.3.1)中 i 为数字符号所在位置的序号。

2. 二进制数

基数为 2,需 2 个数字符号 0,1。

第 i 位的权为 2^i ,计数规律为:逢二进一。

任一个二进制数可表示为

$$(N)_2 = \sum_{i=n-1}^{-m} B_i \times 2^i \quad (B_i \text{ 是 } 0,1 \text{ 中的一个}) \quad (1.3.2)$$

如: $(1010.1)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1}$

3. 八进制数

基数为 8,需 0,1,2,⋯,7 共 8 个数字符号。

第 i 位的权为 8^i ,计数规律为:逢八进一。

对任一个八进制数可表示为

$$(N)_8 = \sum_{i=n-1}^{-m} Q_i \times 8^i \quad (Q_i \text{ 是 } 0,1,2,\dots,7 \text{ 中的一个}) \quad (1.3.3)$$

如 $(502.4)_8 = 5 \times 8^2 + 0 \times 8^1 + 2 \times 8^0 + 4 \times 8^{-1}$

4. 十六进制数

基数为 16,需 0,1,2,⋯,9,A,B,C,D,E,F 共 16 个数字符号。

第 i 位的权为 16^i ,计数规律为:逢十六进一。

对任一个十六进制数可表示为

$$(N)_{16} = \sum_{i=n-1}^{-m} H_i \times 16^i \quad (H_i \text{ 是 } 0,1,\dots,A,B,C,D,E,F \text{ 中的一个})$$

$$\text{如}(\text{FBCD} \cdot \text{E})_{16} = 15 \times 16^3 + 11 \times 16^2 + 12 \times 16^1 + 13 \times 16^0 + 14 \times 16^{-1} \quad (1.3.4)$$

在上述表达式中,对不同数制,我们用附加的下标来区别,其等式右边是按权展开相加的,所以有时也称该多项式为按权展开式。

(二) 数制转换

1. 任意进制数转换成十进制数

对于任意进位计数制,基数可用正整数 x 表示,这时有

$$(N)_x = \sum_{i=n-1}^{-m} K_i \times X^i \quad (K_i \text{ 为第 } i \text{ 位系数,可以为 } 0, 1, \dots, (x-1) \text{ 中任一个}) \quad (1.3.5)$$

X^i : 第 i 位的权

则任意进制的数转换为十进制数只需按上式展开求和即可。

2. 十进制整数转换成任意进制整数

(1) 十进制整数转换成任意进制整数

设 N 为任一个十进制整数,欲将其转换成 n 位 X 进制整数,

$$\begin{aligned} (N)_{10} &= \sum_{i=0}^{n-1} K_i \times X^i \\ &= K_{n-1}X^{n-1} + K_{n-2}X^{n-2} + \dots + K_1X^1 + K_0 \end{aligned}$$

由于等式右边除 K_0 外,其余各项都包含基数 X 的因子,连续除以 X ,则第一次除后的余数是 K_0 ,第二次除后的余数是 K_1 ,...如此下去直至商等于零为止,则最后一次除后的余数为 K_{n-1} 。连续所得的一系列余数 $K_{n-1}, K_{n-2}, \dots, K_1, K_0$,正是要求的 X 进制数的各位。这种方法也常称为“除基取余法”。

例如将 $(15)_{10}$ 转换成二进制数,其过程如下:

$$\begin{array}{rcll} 2 & 15 & & \\ 2 & 7 \cdots 1 & (\text{商 } 7 \text{ 余 } 1) \rightarrow K_0 & \text{低位} \\ 2 & 3 \cdots 1 & (\text{商 } 3 \text{ 余 } 1) \rightarrow K_1 & \\ 2 & 1 \cdots 1 & (\text{商 } 1 \text{ 余 } 1) \rightarrow K_2 & \\ & 0 \cdots 1 & (\text{商 } 0 \text{ 余 } 1) \rightarrow K_3 & \text{高位} \end{array}$$

逆序排列余数便是与该十进制相应的二进制各位的数值。

$$\therefore (15)_{10} = (K_3 K_2 K_1 K_0)_2 = (1111)_2$$

(2) 十进制小数转换成任意进制小数

设 N 为任一个十进制小数,欲将其转换为 m 位 X 进制小数,

$$(N)_{10} = \sum_{i=-m}^{-1} K_i \times X^i = K_{-1}X^{-1} + K_{-2}X^{-2} + \dots + K_{-m}X^{-m}$$

等式两边同乘以基数 X ,可得到整数部分 K_{-1} ,去掉 K_{-1} 后的剩余小数部分再乘以 X ,可得到整数部分 K_{-2} ,...这样由高位向低位逐次进行,直到满足精度要求或乘 X 后的小数部分为 0 为止。这样连乘所得的一系列整数 $K_{-1} K_{-2} \cdots K_{-m}$,正好是所要求的 X 进制小数的各位。这种方法又叫“乘基取整法”。

例如将 $(0.6531)_{10}$ 转换成纯二进制小数。

转换过程如下:

$$\begin{array}{ll}
 0.6531 \times 2 = 1.3062 & K_{-1} = 1 \\
 0.3062 \times 2 = 0.6124 & K_{-2} = 0 \\
 0.6142 \times 2 = 1.2248 & K_{-3} = 1 \\
 0.2248 \times 2 = 0.4496 & K_{-4} = 0 \\
 0.4496 \times 2 = 0.8992 & K_{-5} = 0 \\
 0.8992 \times 2 = 1.7984 & K_{-6} = 1
 \end{array}$$

如只取六位小数能满足精度要求,则可得

$$(0.6531)_{10} = (0.K_{-1}K_{-2}K_{-3}\cdots K_{-m})_2 \approx (0.101001)_2$$

可见,十进制纯小数不一定能转换成完全等值的 X 进制纯小数,这时只要乘到能满足计算精度要求为止。

对于由整数和小数复合而成的十进制数,可以把整数和小数部分分别转换,然后将转换结果组合起来即可。

如将 $(15.6531)_{10}$ 转换成二进制数,

由上例可知: $(15)_{10} = (1111)_2$

$$(0.6531)_{10} \approx (0.10100111)_2$$

则 $(15.6531)_{10} \approx (1111.10100111)_2$

3. 2^n 进制数与二进制之间的转换

2^n 进制数中的 n 大于 2,且为正整数。对于基数是 2 的整数幂的进位制数与二进制数相互转换是十分简便的。其基本方法是:二进制数转换为 2^n 进制数时,首先从小数点部分分别向左和向右把整数及小数每 n 位分成一组,若整数最高位的一组不足 n 位,则在其左边加 0 补足 n 位。若小数最低位的一组不足 n 位,则在其右边加 0 补足 n 位。然后用与每组二进制数所对应的 2^n 进制数取代每组的 n 位二进制数,即得到对应的 2^n 进制数;反之,若将 2^n 进制数转换为二进制数,只需要把相应的 n 位二进制取代每一位 2^n 进制数即可。

例如:把 $(1101001.0100111)_2$ 转换成八进制数方法是:

$$\begin{array}{cccccc}
 (001 & 101 & 001 & \cdot 010 & 011 & 100)_2 \\
 (\downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow) \\
 (1 & 5 & 1 & \cdot 2 & 3 & 4)_8
 \end{array}$$

$$\therefore (1101001.0100111)_2 = (151.234)_8$$

再如,把十六进制数 $(863.6C)_{16}$ 转换成二进制数,则

$$\begin{array}{cccccc}
 (8 & 6 & 3 & \cdot & 6 & C)_{16} \\
 (\downarrow & \downarrow & \downarrow & & \downarrow & \downarrow) \\
 (1000 & 0110 & 0011 & \cdot & 0110 & 1100)_2
 \end{array}$$

$$\text{即: } (863.6C)_{16} = (1000\ 0110\ 0011 \cdot 011011)_2$$

二、带符号数的表示及运算

(一) 有符号二进制的表示方法

前面我们提到的二进制数均没有考虑数的符号,客观存在的二进制数既可能是正数,也可能是负数,那么有符号的二进制数在计算机中如何表示呢?

1. 真值、机器数与原码

把十进制数中正、负数的表示方法运用到二进制数中,即在正数前加“+”号,负数前加“-”号,则+73和-23的二进制数表示形式为:

$$X_1 = +1001001 \quad (+73)_{10}$$

$$X_2 = -0010111 \quad (-23)_{10}$$

然而,在计算机中存储和计算的只能是由“0”“1”代码组成的二进制数,通常用“0”来取代“+”号,用“1”来取代“-”号,则上述两数就可以表示为:

$$X_1 = 01001001 \quad (+73)_{10}$$

$$X_2 = 10010111 \quad (-23)_{10}$$

这样把符号数值化之后,就可用于机器中。这种机器可使用的带符号数就叫做机器数,而机器数所代表的数称为该机器数的真值。当然,除符号数值化外,机器数还有字长一定、运算精度有限及必须规定小数点位置等特点。常用的机器数有原码、补码和反码三种形式。

如果将数 x 的原码记作 $[X]_{\text{原}}$,对机器字长为 n 位的二进制数,其原码一般可表示为

$$[X]_{\text{原}} = \begin{cases} x & 0 \leq x < 2^{n-1} - 1 \\ 2^{n-1} - x & -(2^{n-1} - 1) \leq x \leq 0 \end{cases}$$

由式可见,当 x 为正数时, $[X]_{\text{原}} = x$,仅仅是符号位用“0”取代,这样表示后, x 的原值并未变化。但当 x 为负数时,减 x 实际上变为加上 $|x|$,即 $[X]_{\text{原}} = 2^{n-1} + |x|$,所以只需将 $|x|$ 的数值部分前面的符号位上写成“1”即可。

例如。当机器字长 $n = 8$ 时

$$[+95]_{\text{原}} = \boxed{0} 1011111$$

$$[-95]_{\text{原}} = \boxed{1} 1011111$$

由此看出,原码表示中最高位为符号位,其余 $n - 1$ 位表示数的绝对值。所以原码表示数的范围为 $-(2^{n-1} - 1) \sim +(2^{n-1} - 1)$ 。

对 8 位二进制数,原码表示范围为: $-127 \sim +127$ 。

对 16 位二进制数,原码表示范围为: $-32767 \sim +32767$ 。

这种二进制数的原码表示法简单易懂,但不便于进行加减运算。

2. 反码

如果将数 x 的反码记作 $[x]_{\text{反}}$,对机器字长为 n 位的二进制数,其反码定义如下:

$$[x]_{\text{反}} = \begin{cases} x & 0 \leq x \leq 2^{n-1} - 1 \\ (2^n - 1) + x & -(2^{n-1} - 1) \leq x \leq 0 \end{cases}$$

从定义可看出,对于 n 位二进制数来说,当 x 为正数时, $[x]_{\text{反}}$ 与 x 的区别仅在于符号位用 0 表示;当 x 为负数时, $2^n - 1$ 即为 n 个 1,由于 x 取值范围为: $-(2^{n-1} - 1) \leq x \leq 0$,此时,加 x 实际上变为减 $|x|$,即 $[x]_{\text{反}} = (2^n - 1) - |x|$ 其尾数不会大于 $n - 1$ 个 1,故此时反码表示法中最左边一位是 1。所以反码表示中最左边一位也是符号位。

例如:当机器字长 $n = 8$ 时,

$$[+95] = 01011111$$

已知: $[-95]_{\text{原}} = \boxed{1} 1011111$

则: $[-95]_{\text{反}} = \boxed{1} 0100000$

所以,反码表示数的范围也是 $-(2^{n-1}-1) \sim +(2^{n-1}-1)$,与原码相同。

由上例也可看出,正数的反码等于原码。对于负数而言,知其原码,则保持符号位不变,其余各位变反,就得到其反码。

3. 补码

(1) 补码的概念

根据同余的概念

$$x + Nk = x \pmod{k} \quad N \text{ 为任意整数}$$

在模 k 的意义下,数 x 与该数本身加上其模的任意整数倍之和相等。

当 N 为 1 时,则补数

$$\begin{aligned} [x]_{\text{补数}} &= x + k \pmod{k} \\ &= \begin{cases} x & 0 \leq x < k \\ k + x & -k < x < 0 \end{cases} \end{aligned}$$

由上式确定的两种条件下数 x 的补数,正是补码的定义和补码编码规则的基础。

结合到计算机运算的特点,由于计算机的字长总是有限的,假定其字长为 n ,则计算机中最大的计算范围(包括符号位)为 2^n ,因此计算机中的补码是以 2^n 为模,则数 x 的补码可由下述公式求得:

$$[x]_{\text{补}} = \begin{cases} x & 0 \leq x \leq 2^{n-1} - 1 \\ 2^n + x & -2^{n-1} \leq x < 0 \end{cases} \pmod{2^n}$$

由公式可见,正数的补码等于其原码,负数的补码等于该数加模。所以,严格地说,只有负数才有求补码的问题,有时把这种以 2^n 为模的补码,也称作对 2 的补码。相比之下,反码是以 $2^n - 1$ 为模的补码,所以也称反码为对 1 的补码。

(2) 求补码的方法

1) 根据定义求补码

$$[x]_{\text{补}} = 2^n + x = 2^n - |x|, x < 0$$

如 $x = -1010111$ $n = 8$ 则

$$\begin{aligned} [x]_{\text{补}} &= 2^8 - |-1010111| \\ &= 100000000 - 1010111 \\ &= 10101001 \pmod{2^8} \end{aligned}$$

这种方法因为要作一次减法很不方便,一般不用。

2) 利用反码求补码

由于在 $x < 0$ 时,

$$\begin{aligned} [x]_{\text{补}} &= 2^n + x \\ &= (2^n - 1 + x) + 1 \\ &= [x]_{\text{反}} + 1 \end{aligned}$$

所以若知 x 原码,则求其反码,然后加 1 即得其补码。

$$\begin{array}{rcl}
 \text{如:} & [x]_{\text{原}} = \boxed{1} 0011010 & \\
 & \downarrow & \\
 & [x]_{\text{反}} = \boxed{1} 1100101 & \\
 & \begin{array}{r} +) \\ \hline \end{array} & \begin{array}{r} 1 \\ 1 \end{array} \\
 & [x]_{\text{补}} = 11100110 &
 \end{array}$$

3) 直接求补法

其基本作法是:已知某数 x 的原码,则保持其符号位不变,数值部分从低位向高位逐位进行,在遇到第一个 1 以前,包括第一个 1,按原码照写;第一个 1 后,逐位变反。

$$\begin{array}{rcl}
 \text{例如:} [x]_{\text{原}} = \boxed{1} 1010 \boxed{100} & & \\
 \downarrow & & \\
 [x]_{\text{补}} = \boxed{1} 0101 \boxed{100} & & \\
 \text{不变} \quad \text{变反} \quad \text{不变} & &
 \end{array}$$

(3) 求补与变补

由上述已知,对于正数而言:

$$[x]_{\text{原}} = [x]_{\text{补}} = [x]_{\text{反}}$$

对于负数,只要知其原码,就可方便求得对应的反码和补码。那么已知某数的补码 $[x]_{\text{补}}$,如何求得其原码 $[x]_{\text{原}}$,则有“双补还原”定理:

$$[[x]_{\text{补}}]_{\text{补}} = [x]_{\text{原}} \quad (\text{证明从略})$$

现用实例加以说明。

$$\text{如: } [x_1]_{\text{原}} = 01000011 \quad (+67)$$

$$[x_2]_{\text{原}} = 11000011 \quad (-67)$$

$$\text{则 } [[x_1]_{\text{补}}]_{\text{补}} = [01000011]_{\text{补}} = 01000011 = [x_1]_{\text{原}}$$

$$[[x_2]_{\text{补}}]_{\text{补}} = [10111101]_{\text{补}} = 11000011 = [x_2]_{\text{原}}$$

可见,无论是正数还是负数,其补码的补码等于原码。

但是,假设已知 $[x]_{\text{补}}$,在具体做法上,如果连同其符号位统统变反,然后加 1,结果将得到其相反数的补码 $[-x]_{\text{补}}$,而不会再是 $[x]_{\text{原}}$,为了区别,我们把这种由某数 $[x]_{\text{补}}$,求得其相反数 $[-x]_{\text{补}}$ 的过程叫做变补。

显然求补和变补是两个不同的概念,求补是已知一个数的原码求其补码,而变补是已知某数 $[X]_{\text{补}}$,求其相反数 $[-X]_{\text{补}}$ 。两者在具体操作时,前者是符号位不变,其余各位变反加 1,后者则是连同符号位一起变反加 1,即:

$$[[x]_{\text{补}}]_{\text{求补}} = [x]_{\text{原}}$$

$$[[x]_{\text{补}}]_{\text{变补}} = [-x]_{\text{补}}$$

$$\text{例如: } x = 38 \quad \text{其相反数为 } -38$$

$$\text{则 } [x]_{\text{补}} = [x]_{\text{原}} = 00100110$$

$$\text{若对其变补:} \quad \therefore [x]_{\text{补}} = 00100110$$

$$\begin{array}{r}
 11011001 \\
 +) \quad 1 \\
 \hline
 \therefore [[x]_{\text{补}}]_{\text{变补}} = 11011010
 \end{array}$$