



全国高等教育自学考试指定教材

机械制造与自动化
机电一体化工程
(独立本科段)

工业用微型计算机

附：工业用微型计算机自学考试大纲

课程代码
2241

[2000年版]

编组：全国高等教育自学考试指导委员会
主编：赵长德

机械工业出版社

全国高等教育自学考试指定教材

机电一体化工程 (独立本科段)
机械制造与自动化

工业用微型计算机

(附：工业用微型计算机自学考试大纲)

全国高等教育自学考试指导委员会组编

主编 赵长德

参编 徐海峰

金广业

主审 彭一苇

机械工业出版社

《工业用微型计算机》是机电一体化工程专业(独立本科段)自学考试教材。该书以“工业用微型计算机”为目标,以 IBM PC 计算机为基本机型,讲述了计算机中的数制、微型计算机的发展和微机系统的组成、微处理器等基础知识;8086/8088CPU 的指令系统;汇编语言程序设计和上机操作;微机中的内存与 CPU 的连接;输入输出及其 I/O 接口芯片和应用,模拟量输入输出(D/A, A/D);可编程序控制器(PLC);同时还介绍了 PC 和 PC/104 总线工业控制计算机和计算机网络的发展和应用。

本书内容深入浅出,使读者具备汇编语言程序设计的基本知识和能力,掌握工业用微型计算机常用的接口技术、PLC 的初步应用;了解工控机和网络的发展和应用,为从事工业用微型计算机的开发和应用打下基础。

本书可作为机电类、自控类、电气类各专业本科生和成人教育的教材,有关工程技术人员的参考书。

图书在版编目(CIP)数据

工业用微型计算机/赵长德主编;全国高等教育自学考试指导委员会组编. —北京:机械工业出版社,2000.3

全国高等教育自学考试指定教材

ISBN 978-7-111-07878-4

I. 工… I. ①赵…②全… Ⅰ. 工业控制计算机:微型计算机—高等教育—自学考试—教材 N. TP368

中国版本图书馆 CIP 数据核字(2000)第 03258 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码:100037)

责任编辑:王世刚等 版式设计:冉晓华 责任校对:申春香

北京友谊印刷有限公司印刷

2000 年 3 月第 1 版

2008 年 12 月第 11 次印刷

787mm×1092mm/16·23.875 印张·582 千字

44201-49200

定价:32.00 元

本书如有质量问题,请与当地教材供应部门联系。

组 编 前 言

当您开始阅读本书时，人类已经迈入了二十一世纪。

这是一个变幻难测的世纪，这是一个催人奋进的时代。科学技术飞速发展，知识更替日新月异。希望、困惑、机遇、挑战，随时随地都有可能出现在每一个社会成员的生活之中。抓住机遇，寻求发展，迎接挑战，适应变化的制胜法宝就是学习——依靠自己学习、终生学习。

作为我国高等教育组成部分的自学考试，其职责就是在高等教育这个水平上倡导自学、鼓励自学、帮助自学、推动自学，为每一个自学者铺就成才之路。组织编写供读者学习的教材就是履行这个职责的重要环节。毫无疑问，这种教材应当适合自学，应当有利于学习者掌握、了解新知识、新信息，有利于学习者增强创新意识、培养实践能力、形成自学能力，也有利于学习者学以致用、解决实际工作中所遇到的问题。具有如此特点的书，我们虽然沿用了“教材”这个概念，但它与那种仅供教师讲、学生听，教师不讲、学生不懂，以“教”为中心的教科书相比，已经在内容安排、形式体例、行文风格等方面都大不相同了。希望读者对此有所了解，以便从一开始就树立起依靠自己学习的坚定信念，不断探索适合自己的学习方法，充分利用已有的知识基础和实际工作经验，最大限度地发挥自己的潜能达到学习的目标。

欢迎读者提出意见和建议。

祝每一位读者自学成功。

全国高等教育自学考试指导委员会

1999年7月

编者的话

本书是根据全国考委 1998 年颁发的《关于调整高等教育自学考试机电一体化工程专业考试计划的通知》，将原来的《微机原理与接口技术》课程改为《工业用微型计算机》而编写的。

原来的教材是以 Z80 CPU 为主要内容，早已不能满足当前计算机发展和应用的现状，势在必改。但也应该看到，自微型计算机问世以来，一直以目不暇接的态势飞速发展，新技术、新机型、新应用层出不穷。现在，又出现了全球数字化、信息化、网络化的大趋势。所以，计算机的教学内容也需要不断更新和充实！但是要使教材跟上微型机这种更新换代的步伐，实际上又是不可能的，只能做到相对稳定。

计算机教育应该是层次教育，第一层次是计算机应用基础，第二层次是高级语言程序设计和软件基础，第三层次是微机的硬件基础，第四个层次是计算机在专业中的应用，前两个层次已经在大专和本科专业的软件技术基础课程中进行。本门课程处于第三个层次，是机电类考生学习和掌握计算机硬件知识和汇编语言程序设计的一门重要课程。课程的任务是使读者从理论和实践上掌握微型机的组成、工作原理、接口电路，建立微机系统的整体概念，以使读者具备微机系统软硬件开发的初步能力。尽管当前计算机技术发展非常快，CPU 已经用到 Pentium-III，但主流机型仍然是 IBM PC。所以按照本门课程的定位，仍以 IBM PC/XT 或 AT 为背景机，以“工业用微型计算机”为目标，讲述微机系统的组成、工作原理、接口芯片和典型应用，在此基础上适当介绍在机电一体化系统中广泛应用的工业控制计算机，特别是 PC/104 总线嵌入式计算机，但又不是了一本工控机的专著。为了与以前开考的课名称相区别，叫做《工业用微型计算机》。通过本书的学习和上机实践，培养读者分析问题和解决问题的能力，以适应机电一体化工程的发展和社会的需要。

在内容的取舍方面，着重从非计算机专业的特点出发，注重应用，叙述力求深入浅出，从外部特性讲述芯片的功能，尽量多举应用实例。由于微型计算机是一门实践性非常强的课程，除了学习教材的内容，还要大力加强实践环节，通过大量的软件和硬件上机实验，才能够理解书本内容，提高实际动手和应用能力。

全书由清华大学精密仪器系赵长德教授编写和定稿，作者多年从事机电类微机系统原理与应用的教学工作；第五、六章由多年从事计算机控制技术的清华大学精密仪器系徐海峥讲师编写；第七章由多年从事 PLC 应用的奥特曼公司金广业高级工程师编写。

本书的编写得到全国高等教育自学考试机械类专业委员会秘书长、西安交通大学杨林森教授的指导和支持。北京理工大学计算机系彭一苇教授在百忙中主持了审稿工作；北京理工大学机械工程学院焦振学教授和清华大学同方公司杨春武高级工程师也参加了审稿，他们认真仔细地审阅了书稿，提出了宝贵的修改意见，在此表示深深的感谢。本书的编写和出版，还得到自学考试办公室教材处、我所在的清华—盛博嵌入式计算机联合实验室的支持，还有我的研究生温小刚同学搜集了有关网络的资料，写出第八章第二节的初稿，在此一并致谢。

由于编者水平的限制，加之时间仓促，书中一定存在错误或不妥之处，请读者不吝指正。

编者

1999 年 12 月

目 录

编者的话

第一章 微型计算机基础 1

第一节 微型计算机的发展 1

第二节 计算机的数制和编码系统 3

一、无符号数的表示和运算 3

二、带符号数的表示方法 7

三、8 位与 16 位二进制的表示范围 9

四、计算机中的字符编码 10

第三节 微型计算机系统的组成 11

一、微型计算机系统的构成 11

二、工业微型计算机概况 12

三、微型计算机的外围设备 15

第四节 微处理器 15

一、Intel 8086/8088 微处理器 15

二、Intel 80286、80386、80486 微处理器 22

三、Pentium 微处理器 26

四、CPU 主板和 CPU 卡 27

习题 29

第二章 8086/8088 微处理器的指令

系统 31

第一节 寻址方式 31

一、指令格式 31

二、立即寻址 31

三、寄存器寻址 32

四、直接寻址 32

五、寄存器间接寻址 32

六、变址寻址 33

七、基址寻址 34

八、基址一变址寻址 34

第二节 8086/8088 的指令系统 36

一、数据传送指令 37

二、算术运算指令 45

三、逻辑运算和移位指令 53

四、串操作指令 61

五、控制转移指令 66

六、处理器控制指令 74

第三节 DEBUG 调试程序的简单应用 75

一、DEBUG 调试程序的功能 75

二、利用 DEBUG 学习寻址方式 80

三、利用 DEBUG 调试程序段 81

习题 83

第三章 汇编语言程序设计 84

第一节 程序设计语言概述 84

一、机器语言 84

二、汇编语言 84

三、高级语言 84

第二节 汇编语言源程序的格式 85

一、分段结构 86

二、名字 86

三、助记符和伪操作 87

四、操作数 87

五、注释 91

六、汇编语言程序结构 91

第三节 常用伪操作 93

一、段定义伪操作 93

二、数据定义伪操作 96

三、符号定义伪操作 97

四、过程定义伪操作 98

五、模块定义与连接伪操作 99

六、宏处理伪操作 101

第四节 汇编语言程序设计 105

一、程序设计步骤 105

二、循环与分支程序设计 106

三、子程序设计 108

四、DOS 和 BIOS 调用 112

五、常用汇编语言程序设计举例 119

第五节 汇编语言的上机过程 139

一、工作环境 139

二、建立汇编语言源文件 139

三、MASM 和 LINK 程序的应用 140

习题 141

第四章 微型计算机的内存及其与 CPU 的

连接 143

第一节 概述	143	三、可编程序中断控制器 Intel	
一、存储器的分类	143	8259A	187
二、半导体存储器的分类	143	四、8259A 的编程	192
三、半导体存储器的指标	144	第四节 计数器/定时器电路	202
第二节 随机读写存储器 (RAM)	145	一、可编程计数/定时器 Intel8253	202
一、静态 RAM	145	二、8253 的应用	209
二、动态 RAM	146	第五节 并行 I/O 接口	212
第三节 只读存储器 (ROM)	151	一、可编程并行 I/O 接口芯片	
一、掩膜 ROM	151	Intel 8255A	212
二、可擦可编程只读存储器		二、8255A 的应用	215
(EPROM)	151	第六节 异步串行 I/O	220
三、电可擦除可编程 ROM		一、异步串行通信规程和接口标准	220
(EEPROM)	154	二、可编程串行接口芯片 8250	222
第四节 CPU 与存储器的连接	155	三、8250 的编程	225
一、连接时应注意的问题	155	四、8250 的应用	227
二、典型 CPU 与存储器的连接	156	五、8251A 串行接口芯片及其应用	231
第五节 IBMPC/XT 中的存储器	157	习题	235
一、存储空间的分配	157	第六章 模拟量输入输出与数据采	
二、ROM 子系统	158	集卡	237
三、RAM 子系统	159	第一节 数模转换器 D/A	237
第六节 扩展存储器及其管理	160	一、概述	237
一、寻址范围	160	二、D/A 转换器的工作原理	237
二、存储器管理	161	三、D/A 转换器的主要技术指标	240
三、高速缓存器	164	四、典型 D/A 转换器芯片	240
四、存储器管理软件	165	五、D/A 转换器与总线的接口	243
习题	167	第二节 模数转换器 A/D	247
第五章 输入输出及接口芯片的		一、A/D 转换器工作原理	247
应用	168	二、A/D 转换器的主要技术性能	248
第一节 概述	168	三、典型 A/D 转换器芯片	249
一、I/O 接口	168	四、A/D 转换器与总线接口	252
二、I/O 的传送方式	169	第三节 多路开关	253
三、I/O 端口的寻址方式	169	一、数据采集系统对多路开关的要求	253
四、I/O 的控制方式	170	二、几种多路开关芯片	254
第二节 PC/AT 总线及简单 I/O 接口	173	三、多路开关的主要技术参数	255
一、PC/XT 总线和 PC/AT 总线	174	第四节 采样保持器	256
二、PCI 总线	178	一、采样保持器的工作原理	256
三、I/O 端口地址	182	二、常用的采样保持器集成芯片	257
四、简单 I/O 接口及 I/O 卡	183	第五节 数据采集卡及其应用	258
五、LED 显示接口	183	一、PCL-818 模块主要技术性能	258
第三节 中断	185	二、电路组成	259
一、中断的概念, IBM-PC 的中断		三、编程	260
系统	185	四、PCL-818 应用举例	260
二、中断响应	186	习题	261

第七章 可编程序控制器	262	五、可编程序控制器在机电一体化中的	
第一节 可编程序控制器的发展状况	262	应用	300
一、可编程序控制器的发展历史	262	习题	302
二、可编程序控制器的发展现状	263	第八章 微型计算机的发展和应用	304
三、可编程序控制器的标准	264	第一节 工业控制计算机	304
第二节 可编程序控制器的工作原理	264	一、工业控制计算机	304
一、可编程序控制器的基本组成	264	二、PC/104 总线嵌入式工控机	307
二、可编程序控制器的输入输出模板	266	三、单片计算机的发展和应用	313
三、编程语言和程序结构	271	第二节 计算机网络与 Internet 国际互	
四、输入输出映像区	272	联网	315
五、周期扫描机制	272	一、网络技术发展概述	315
第三节 可编程序控制器的指令系统	275	二、网络体系结构与网络协议	317
一、逻辑操作指令	275	三、Internet 国际互联网	321
二、定时器和计数器指令	278	四、工业控制网络	330
三、基本运算指令	282	习题	332
四、数据处理指令	284	附录	333
五、程序控制指令	287	参考文献	347
第四节 可编程序控制器的应用	290	后记	348
一、可编程序控制器系统的硬件选择	290	《工业用微型计算机》自学考试大纲	349
二、可编程序控制器的程序设计方法	293	后记	373
三、可编程序控制器系统的故障处理	295		
四、典型的可编程序控制器系统结构	298		

第一章 微型计算机基础

本章主要说明微型计算机的发展、计算机中的数制和编码、微型计算机的组成，以及典型的微处理器特点、系统主机板的配置。

本章的重点是第二和第四节，读者应掌握计算机中无符号数和有符号数的表示方法，各种数制的转换关系，8086/8088 CPU 中的功能结构和各个寄存器的用途、存储器的结构，进而了解计算机的工作原理，明确本门课程的学习内容和目的。

第一节 微型计算机的发展

自 1946 年世界上第一台电子计算机问世以来，计算机科学与技术获得了日新月异的飞速发展，它的应用已经深入到人类社会生活的各个方面。特别是在跨进 21 世纪的今天，人们都在谈论“科教兴国”、“信息时代”和“知识经济”，而且意识到：计算机和网络的知识已经成为当代大学生知识结构中不可缺少的有机组成部分。

计算机按其性能、价格和体积可分为：巨型机、大型机、中型机、小型机和微型机。微型机是 20 世纪 70 年代研制成功的，一方面是由于当时军事、工业自动化技术的发展，需要体积小、功耗低、可靠性好的微型计算机；另一方面，大规模集成电路技术的最新成果也为微型机的产生打下了坚实的物质基础。

微处理器是微型机的核心芯片，通常简称为 MP (Micro Processor)，它是将计算机中的运算器和控制器集成在一个硅片上制作的集成电路。这样的芯片也被称为中央处理单元，一般简称为 CPU (Central Processing Unit)。

微型计算机简称 MC (Microcomputer)，它是由微处理器、一定数量的内存和 I/O 接口电路以及必要的键盘、显示器等外部设备组成的计算机。

近 20 多年以来，微处理器和微型计算机得到极快的发展，几乎每两年微处理器的集成度就翻一番，每 2~4 年就换一种新的型号，更新换代的速度使人们追赶不及。

第一代 4 位和低档 8 位微处理器(1971~1973 年) 美国 Intel 公司率先在 1971 年试制了 4 位微处理器 4004，其集成度只有 2000 个晶体管。次年又推出 8008 微处理器，这是低档 8 位的微处理器。

第二代中高档 8 位微处理器(1974~1978) 众多半导体公司纷纷加入研制微处理器的行列，逐步形成了以 Intel、Motorola、Zilog 三个公司的产品为代表的不同系列的微处理器。1974~1975 年，推出中档微处理器，1976~1978 年，出现高档微处理器。例如，在我国应用达十年之久的 Z80 就是 Zilog 的产品，而 Intel 的 8085 和 Motorola 的 MC6809 也是这一时期推出的代表性产品。它们比第一代产品有很多改进，集成度提高 2~4 倍，运算速度提高 10~15 倍，指令系统更完善了，具有典型的计算机体系结构以及中断、DMA 功能。计算机软件除汇编语言外，还增加了 BASIC、FORTRAN，以及 PL/M 等高级语言。当时出现了具有磁盘操作系统的微型计算机，如以 MC6502 为 CPU 的苹果机。

第三代 16 位微处理器(1978~1981 年) 随着一个硅片可集成一万只以上晶体管的超大规模集成电路的研制成功, 出现了 16kb (bit) 和 64kb 的半导体存储器。同时, 微处理器发展为第三代: Intel 8086 CPU 的集成度达到 29000 个晶体管, Z8000 为 17500 个晶体管, MC68000 为 68000 个晶体管。从各项性能指标看比上一代产品提高很多。例如, 平均指令时间约为 $0.15\mu\text{s}$, 字长为 16 位, 指令系统更加丰富, 具有多级中断。还配备了功能较强的系统软件, 为与原 8 位机相衔接, 方便 8 位机用户, Intel 公司又推出了 8088 CPU, 其指令系统完全与 8086 兼容, 内部仍然为 16 位的功能结构, 而外部数据总线是 8 位。IBM 公司以 8088 CPU 组成了微型机 IBM PC 和 IBM PC/XT, 并且采用新的策略——公布了该计算机的全部资料, 使许多厂商为其研制配套的外设产品。由于其性能价格比好, 迅速占领了国际市场。进一步, Intel 公司又推出性能优良的 80286CPU 和协处理器 80287, 以此为基础 IBM 公司推出了 IBM PC/AT 计算机。同时, 微型机的应用也逐渐普及开来。从此, 真正进入了个人计算机的时代。

第四代 32 位高档微处理器(1985~1993 年) 1985 年, 推出了 32 位微处理器 80386DX, 它的数据总线和地址总线都是 32 位, 采用 80387 做协处理器。1990 年, 该公司又在 386 的基础上研制出新一代微处理器 80486。它相当于把 80386 CPU 和 80387 协处理器以及 8KB ($8 \times 1024\text{Byte}$) 的高速缓冲存储器全部集成在一块芯片上, 性能比 80386 大为提高。

第五代 64 位高档微处理器 1993 年, Intel 公司研制出新一代奔腾 (Pentium) 微处理器, 它的外部数据总线为 64 位, 工作频率 66MHz 以上, 以它为 CPU 的微型机称为 586 机或奔腾计算机。IBM、APPLE、Motorola 几个公司合作生产的 Power PC 芯片又是一种性能优异的 64 位微处理器芯片, 以它为 CPU 的微型机叫 Macintosh。

以后 Intel 又推出 Pentium - II 微处理器。Advanced Micro Device 公司 (简称 AMD) 的 K6 是与 Pentium - II 性能相当的 CPU, 这些公司竞争的结果, 使 CPU 的价格不断降低、性能不断提高, 而用户受益。到了 1999 年, Intel 公司展示了 Pentium - III 型微处理器。Intel 批量生产这种高速芯片, 用作高档 PC 机的中央处理单元, 除主频提高到近千 MHz 外, 其强大的网络功能也令人耳目一新。

AMD 公司也紧跟 Intel 公司, 推出 K6 系列的最新处理器 AMD - K6 - 3, 它将主要针对高端用户和商业用户。该处理器具有优秀的 3D 动画性能, 采用 $0.25\mu\text{m}$ 处理工艺制造, 芯片上集成有高速 256K L2 缓存, 超标量 MUX 技术, 在 118mm^2 的面积集成有 2130 万个晶体管。由于拥有片上高速 L2 缓存和外部 L3 缓存, 性能得以大幅度提高。总之微处理器在竞争中发展, 技术不断进步, 其操作系统也从 DOS 发展到 WINDOWS、LINUX, 使微型机的性能不断提高。

应该说, 我国在微处理器的研制还有相当差距, 但利用国际市场上的微处理器及其配套芯片, 由联想等公司制造的各种品牌的计算机, 其性能也达到了国际水平, 市场占有率正在扩大。由中科院软件所等单位研制出的“红旗 LINUX 中文操作系统”, 是拥有我国知识产权的中文操作系统, 对今后计算机应用的推广, 特别是网络安全具有重要意义。

现代计算机向着多媒体和网络化发展, 一台计算机上安装声卡、话筒、图像处理设备, 光盘驱动器等硬件, 再配备相应的系统软件, 就可以组成具有文字、语音输入、图形和图像处理能力的多媒体电脑。

在机电一体化系统和产品中广泛使用着工业控制计算机, 其中包括 PC 总线和 PC/104 总

线工控机、单片机以及可编程控制器 (PLC) 都发挥各自的优势, 在应用中不断发展。

计算机网络的发展使计算机的应用又推向了新的里程碑, 使全球信息化成为可能。这几年 Internet 国际互连网的发展, 使得社会生活的各个方面都发生了深刻的变化。例如人们通过网络收发电子邮件 (Email), 通过网络购物 (电子商务), 通过网络打电话 (IP 电话), 通过万维网 (WWW) 浏览政府、企业、报刊的各种信息。通过网络听音乐、学习、远程医疗等。这是真正地进入了数字化、信息化的时代。网络的发展和应用前景鼓舞着人们更加努力地学习计算机知识, 提高应用计算机的能力和水平。

第二节 计算机的数制和编码系统

本节重点介绍进位计数制, 不同进位计数制之间的相互转换以及各种常用的编码系统。

一、无符号数的表示和运算

(一) 进位计数制

人们在日常生活中, 采用多种进制的数字系统。最常用的是十进制。例如

$$1998 = 1 \times 10^3 + 9 \times 10^2 + 9 \times 10^1 + 8 \times 10^0$$

任何一个十进制数都可表示为

$$X = x_m \times 10^m + \cdots + x_0 + x_{-1} \times 10^{-1} + \cdots + x_{-n} \times 10^{-n} = \sum_{i=-n}^m x_i \times 10^i \quad (2-1)$$

其中, 10 称为十进制数的基数, 而所在位数 i 的权为 10^i 。对任意进制来说, 所谓基数 J , 就是用来选用的不同数制的个数。所以, 十进制的 J 就是 10, 可以选用的数 x_i 为 0~9 共十个数字, 且逢十进一。而小数点左边的权依次为 $J^0, J^1, \dots, J^m$; 小数点右面的权依次为 $J^{-1}, J^{-2}, \dots, J^{-n}$ 。而且, 小数点左移一位等于减小 J 分之一, 小数点右移一位等于增大 J 倍。如果在计算机内要使用十进制数时, 每位数就要用 0~9 共十个数字, 这在电路上无法实现。目前, 一位数字电路仅有两个不同的稳定状态, 且可相互转换, 可表示一位二进制数。所以, 在计算机内都使用二进制。其特点是: 基数为 2, 需要两个数字符号 0 和 1, 逢二进一。对任何二进制数 X 都可展开为

$$X = x_m \times 2^m + \cdots + x_0 + x_{-1} \times 2^{-1} + \cdots + x_{-n} \times 2^{-n} = \sum_{i=-n}^m x_i \times 2^i \quad (2-2)$$

第 i 位数是 0 或 1, 对应的数值应为 $x_i \times 2^i$, 并称 2^i 为第 i 位的权; 例如二进制数

$$1101.1011 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4}$$

因为使用二进制需用较多的位数, 书写很繁, 所以在书写时还经常使用十六进制, 它的特点是: 基数为 16, 使用 16 个数字符号, 它们是 0~9, A, B, C, D, E, F; 对于十六进制的加法运算是逢 16 进位。任何一个十六进制数都可表示为

$$x = x_m \times 16^m + \cdots + x_0 \times 16^0 + x_{-1} \times 16^{-1} + \cdots + x_{-n} \times 16^{-n} = \sum_{i=-n}^m x_i \times 16^i \quad (2-3)$$

第 i 位数是上述 16 个符号之一, 对应的数值为 $x_i \times 16^i$, 并称 16^i 为第 i 位的权。为了避免各种进位计数制的混乱, 习惯用后缀表明数的进制:

后缀 B 表示二进制; 后缀 H 表示十进制; 后缀 D 表示十进制 (也可不加后缀)。

例如: 10011011B—是二进制数; 9BH—是十六进制数; 155D—是十进制数; 这些数都表示同一数值, 即十进制的 155, 只是使用的进制不同而已。

(二) 各种进制数的转换

尽管有不同的进制，但在计算机中的数只能用二进制数，十六进制只是适于读写方便的需要，而十进制则是人们日常生活所必需的。因此，就要掌握各种进制数的转换关系。由于 $2^4 = 16$ ，一位十六进制数可用 4 位二进制表示。它们之间存在着唯一而又直接的对应关系，如表 1-1 所示，可见二进制和十六进制之间的转换是十分简单的。

表 1-1 二进制、十进制、十六进制数对照表

十进制数	十六进制数	二进制数
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	10000

1. 二进制和十六进制间的相互转换

(1) 十六进制转换为二进制数 不论是十六进制的整数还是小数，只要把每一位十六进制的数用相应的二进制数来代替，就可以转换为二进制。例如

$$\begin{array}{cccc}
 9 & B & A & 6 \\
 \downarrow & \downarrow & \downarrow & \downarrow \\
 1001 & 1011 & 1010 & 0110 \\
 \text{即 } 9B.A6H = 10011011.1010011B
 \end{array}$$

(2) 二进制转换为十六进制这种转换 可分两步进行：对整部分，从小数点向左数每 4 位二进制为一组，最后不足的前面补零。对小数部分，从小数点向右数，每 4 位一组，最后不足四位的后面补 0，然后把每 4 位二进制数用相应的十六进制数代替，即可转换为十六进制数。例如

$$\begin{array}{cccc}
 1011 & 0111.0101 & 0100 & \\
 \downarrow & \downarrow & \downarrow & \downarrow \\
 B & 7 & 5 & 4 \\
 \text{即 } 10110111.010101B = B7.54H
 \end{array}$$

2. 二进制、十六进制、十进制的相互转换

(1) 十六进制转换为十进制数 对给定的十六进制数，只要按前述式 (2-3) 展开，就可得到对应的二进制数。例如

$$\begin{aligned}
 ABCDH &= 10 \times 16^3 + 11 \times 16^2 + 12 \times 16^1 + 13 \times 16^0 \\
 &= 10 \times 4096 + 11 \times 256 + 12 \times 16 + 13 \\
 &= 43918D
 \end{aligned}$$

八位二进制数称为一个字节，若用十六进制表示，其范围为 00H~FFH。而 FFH = 255。16 位二进制数（2 字节）可用 4 位十六进制数表示，其最大的数为 FFFFH = 65535D

(2) 二进制数转换为十进制数 对所给的二进制数，只要按前述的式 (2-2) 展开，即可得到对应的十进制数。例如

$$1011.1010B = 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-3} = 11.625$$

对给定的一个字节或两个字节的二进制数也可先化成十六进制数，再转换成十进制数。例如

$$\begin{aligned}
 1011101001110101B &= BA75H \\
 &= 11 \times 16^3 + 10 \times 16^2 + 7 \times 16^1 + 5 \times 16^0 = 47733D
 \end{aligned}$$

(3) 十进制整数转换为二进制数 把十进制整数转换为二进制数，一般采用除 2 取余法。例如

$$215D = x_m \times 2^m + \dots + x_0 \times 2^0$$

只要决定 $x_m, x_{m-1}, \dots, x_0$ 的值，就可写出二进制数，因 $2^0 = 1$ ，所以 $(215 - 2^0)$ 一定是 2 的整数倍， $215 \div 2$ 所得的余数即为 x_0 。其转换过程为

$$\begin{aligned}
 215 \div 2 &= 107 && (\text{商}), \text{余数} = 1 = x_0; \\
 107 \div 2 &= 53 && (\text{商}), \text{余数} = 1 = x_1; \\
 53 \div 2 &= 26 && (\text{商}), \text{余数} = 1 = x_2; \\
 26 \div 2 &= 13 && (\text{商}), \text{余数} = 0 = x_3; \\
 13 \div 2 &= 6 && (\text{商}), \text{余数} = 1 = x_4; \\
 6 \div 2 &= 3 && (\text{商}), \text{余数} = 0 = x_5; \\
 3 \div 2 &= 1 && (\text{商}), \text{余数} = 1 = x_6; \\
 1 \div 2 &= 0 && (\text{商}), \text{余数} = 1, \text{商为 } 0, \text{转换结束。}
 \end{aligned}$$

故 $215D = 11010111B$ 。

(4) 十进制整数转换为十六进制数 同转换为二进制数的道理一样，也可采用除 16 取余例如 215D 转换为十六进制的过程为

$$\begin{aligned}
 215 \div 16 &= 13 && (\text{商}), \text{余数} = 7 = x_0; \\
 13 \div 16 &= 0 && (\text{商}), \text{余数} = 13 = x_1; \text{商为 } 0, \text{转换结束。}
 \end{aligned}$$

故 $215D = D7H$ ；通常写成 0D7H，D 前面的 0 字说明 D 不是英文字符 D 而是数字 13。又如 12345D 的转换过程为

$$\begin{aligned}
 12345 \div 16 &= 771 && (\text{商}), \text{余数} = 9 = x_0; \\
 771 \div 16 &= 48 && (\text{商}), \text{余数} = 3 = x_1; \\
 48 \div 16 &= 3 && (\text{商}), \text{余数} = 0 = x_2; \\
 3 \div 16 &= 0 && (\text{商}), \text{余数} = 3 = x_3, \text{商为 } 0, \text{结束。}
 \end{aligned}$$

故 $12345D = 3039H$ ，然后可化成二进制数 0011000000111001B。

(5) 十进制纯小数转换成二进制小数 采用乘 2 取整法, 设 x 是十进制数的小数部分, 步骤为

〈1〉取 $i = -1$, x 作为被乘数;

〈2〉用 2 乘 x 。取整数部分为 x_i , 取其小数部分为新的被乘数 x_i ;

〈3〉使 i 减 1;

〈4〉重复 〈2〉. 〈3〉步骤直到小数部分为零, 则结束转换。但也有可能小数部分不为零, 可根据计算机的字长或精度要求, 在适当的时候结束转换。

例如, 把 0.625 转换成二进制数

2×0.625	
1.25	整数部分 = 1 = x_{-1}
2×0.25	取小数部分为新被乘数
0.50	整数部分 = 0 = x_{-2}
2×0.50	取小数部分为新被乘数
1.0	整数部分 = 1 = x_{-3}
0.00	小数部分为 0, 转换结束.

故 $0.625 = 0.101\text{B}$.

分别把整数部分和小数部分转换为二进制, 然后合并就可将一个任意十进制数转换成二进制数。例如

$$215.625 = 11010111.101\text{B}$$

(三) 二进制数的加法规则:

二进制的运算规则与十进制类似, 但因二进制数只有 0 和 1 两个数, 故运算规则比十进制要简单的多。

(1) 加法运算 二进制的加法规则

〈1〉 $0+0=0$; 〈2〉 $0+1=1+0=1$; 〈3〉 $1+1=10$ 有进位; 〈4〉 $1+1+1=11$ 有进位

(2) 减法运算 二进制的减法规则

〈1〉 $0-0=0$; 〈2〉 $1-0=1$; 〈3〉 $1-1=0$; 〈4〉 $0-1=1$ 有借位

(3) 乘法运算 二进制的乘法规则为

〈1〉 $0 \times 0 = 0$; 〈2〉 $0 \times 1 = 0$; 〈3〉 $1 \times 0 = 0$; 〈4〉 $1 \times 1 = 1$

(4) 除法运算 与十进制除法类似, 步骤如下:

①从被乘数最高位开始查找, 找到大于除数的位数, 找到这一位时, 商记为一, 并将选定的被乘数减除数, 得余数; ②然后把被乘数的下一位移到余数上, 将余数减除数, 若够减, 商为一; 若不够减, 商为零; ③重复步骤②, 直至把被乘数的所有位都下移完为止。

$$\begin{array}{r} 1001 \\ 110 \overline{) 111011} \end{array}$$

结果: 商为 1001, 余数为 101。继续往下除, 商为小数, 若除不尽, 根据精度要求, 取足位数即可。

(5) “与”运算 规则为按位进行“与”运算; 进行“与”操作的两位均为一, 这“与”的结果为 1; 两位中有一位为 0, 则“与”的结果为 0。“与”运算又称逻辑乘, 一般用符号“ $\wedge$ ”或“ $\cdot$ ”来表示。

11011001

∧ 00110100

00010000

(6) “或”运算规则 是按位进行“或”运算；进行“或”操作的两位中有一位为1，则“或”的结果为1 两位均为0，则“或”的结果为0。“或”运算又称逻辑加，一般用符号“V”或“+”来表示。

(7) “异或”运算规则 为按位进行“异或”运算，进行“异或”操作的两位不同时（即一位为1，另一位为0），则“异或”的结果为1，两位相同时（即都为1，或都为0），则“异或”结果为0。“异或”运算一般用符号“ $\oplus$ ”表示。

二、带符号数的表示方法

(一) 机器数的概念

前面讲的二进制，没有提到符号问题，实际上是一种无符号数的表示。但在计算机中，显然可能有正数和负数。通常一个有符号数的最高位为符号位，即数的符号在机器中也数码化了，我们把一个数放在计算机中的表示形式叫机器数，而这个数本身就称为这个机器数的真值。一个有符号数，由于编码不同，也有几种机器数；反之，一个机器数，由于编码方法不同，又可代表几种真值，如表1-2所示。

表 1-2 数的表示法

机器数		真 值 (十进制)			
二进制数码	十六进制表示	作无符号数	作原码	作反码	作补码
00000000	00	0	+0	+0	+0
00000001	01	1	+1	+1	+1
⋮	⋮	⋮	⋮	⋮	⋮
01111110	7E	126	+126	+126	+126
01111111	7F	127	+127	+127	+127
10000000	80	128	-0	-127	-127
10000001	81	129	-1	-126	-126
⋮	⋮	⋮	⋮	⋮	⋮
11111110	FE	254	-126	-1	-1
11111111	FF	255	-127	-0	-0

更广义的说，在计算机内（存在内存或寄存器中）的数就是机器数，它可以代表无符号数，也可以代表有符号数，有时还可以代表字符，它究竟代表什么由编程者确定，下面通过一些实例理解机器数的真正概念。

(二) 原码

如上所示，正数的符号位用零表示，负数的符号位用1表示，符号位之后表示数值的大小这种表示方法称为原码。例如：

$$x = +114, [x]_{\text{原}} = 01110010\text{B}$$

$$x = -114, [x]_{\text{原}} = 11110010\text{B}$$

这里，后面的7位表示数值部分，最高位为符号位。如果字长为16位二进制时，那后面的15位为数值部分，最高位为符号位。原码的表示方法很简单，缺点是原码的“0”有两个：

$$[+0]_{\text{原}} = 00000000\text{B}$$

$$[-0]_{\text{原}} = 10000000\text{B}$$

另外, 当两个异号数相加, 就要做减法, 为了把减法运算改为加法运算, 引进了反码和补码。

(三) 反码

正数的反码与原码相同。最高位一定为 0, 代表符号, 其余位为数值位。负数的反码其符号位为 1, 与原码相同, 数值位是将其负数的原码的数值位按位取反。例如:

$$x = -4, [x]_{\text{反}} = 11111011\text{B}$$

$$x = -0, [x]_{\text{反}} = 11111111\text{B}$$

$$x = -127, [x]_{\text{反}} = 10000000\text{B}$$

显然, 反码的 0 也有 2 个, $X = +0, [x]_{\text{反}} = 00000000\text{B}$

(四) 补码

正数的补码表示与原码相同, 即最高位为符号位, 用“0”表示, 其余位为数值位。而负数的补码为其反码加 1, 即在反码的最低位加 1 形成补码。例如:

$$x = -4, [x]_{\text{补}} = [x]_{\text{反}} + 1 = 11111011\text{B} + 1 = 11111100\text{B} = \text{FCH}$$

$$x = -127, [x]_{\text{补}} = [x]_{\text{反}} + 1 = 10000001\text{B}$$

$$x = -0, [x]_{\text{补}} = [x]_{\text{反}} + 1 = 00000000\text{B}$$

表 1-2 中给出了不同的机器数, 当它作为无符号数、原码、反码、补码时所代表的真值。8 位二进制补码有以下特点:

$$(1) [+0]_{\text{补}} = [-0]_{\text{补}} = 00000000\text{B};$$

$$(2) 8 \text{ 位二进制补码所能表示的数值为 } +127 \sim -128;$$

(3) 当求补码的真值时, 如最高位为 0 时, 其余 7 位即为此数的二进制值。但高位为“1”时 (即负数), 需把其余 7 位求反以后最低位加 1, 才是它的二进制值。例如:

$$[x]_{\text{补}} = 10010100\text{B}$$

$$x = -(1101011 + 1) = -1101100\text{B} = -108\text{D}$$

(4) 当负数采用补码时, 就可把减法转换为加法。例如:

$$x = 24 - 10 = 14$$

也可用

$$[x]_{\text{补}} = [24]_{\text{补}} - [10]_{\text{补}}$$

而

$$[24]_{\text{补}} = 00011000\text{B} \quad 00011000$$

$$[-10]_{\text{补}} = 11110110\text{B} \quad + 11110110$$

$$\hline 100001110$$

↓

自然丢失

故 $[x]_{\text{补}} = 00001110\text{B}$, 即 $x = 14$, 同减法结果一致。

在日常生活中, 有不少补数的例子, 如手表, 若标准时间为 7 点, 而现在表的实际指示为 9 点, 要拨到 7 点。一种方法是倒拨 2 小时, 另一种是顺拨 10 小时, 这里的顺拨 (+10) 与倒拨 (-2) 对模 12 互为补数, 即 $9 - 2 = 9 + 10 \pmod{12}$, 其意思是, 等号两边同除以 12, 它们的余数相同, 即共模同余, 其中 12 称为模 (Mod 12)。

根据同样原理,可得出十进制数的补码。例如一个圆形两位十进制里程表,初始位置为0,里程表顺时针转一周时为100km,此时又回到初始位置。当往前走1km时里程表指示为01,而往回倒走1km时里程表指示为99,所以99表示-1。或者说-1的补码是99,其模为100。

同理,采用8位二进制时,其模为100H,因此-1的补码是 $(100H - 01H) = FFH$,十进制数-10的补码是 $(100H - 0AH) = F6H$,...,直到-128的补码是 $(100H - 80H) = 80H$ 。这可使求补码的计算速度加快。现在举例说明原码、补码、反码的求法,例如已知真值(十进制数),求其反码和补码。这些十进正数制为+126、-126、+110、-110。

把十进制正数转换成二进制数可用除2取余法,但直接变成16进制更快,书写也简单,上述四个数均在-128~+127之内,都可用8位二进制表示。先求原码+126=7EH,而-126的原码应为 $80H + 7EH = FEH$,因为只需在最高位 D_7 位改为1。+110=6EH,故-110的原码为EEH,也是在符号位(D_7 位)改为1。再求反码、补码。但对于正数来说,其反码=原码=补码。而对于负数来说,可用对应的符号位不动,其余各位按位求反,即可得反码,反码加1即得补码。故-126的反码为81H,补码为82H,而-110的反码为91H,补码为92H。另一种求补码的办法更为简单,例如-1的补码为FFH,即 $100H - 01H = FFH$ 。注意,减1时不够减向高位借1为低位的16,所以求-126的补码可用100H减去+126的原码7EH得到,即 $100H - 7EH = 82H$,再求反码 $82H - 1 = 81H$,可将其结果列表如表1-3所示。

表 1-3 十进制数的原码、反码和补码

真值	+126		-126		+110		-110	
原码	01111110	7EH	11111110	FEH	01101110	6EH	11101110	EEH
反码	01111110	7EH	10000001	81H	01101110	6EH	10010001	91H
补码	01111110	7EH	10000010	82H	01101110	6EH	100010010	92H

三、8位与16位二进制数的表示范围

(一) 8位二进制的范围

1) 无符号数 0~255 (或用0~FFH表示)

2) 有符号数 (1) 原码 最小 11111111B→-127

最大 01111111B→+127

两个零 $\begin{cases} 10000000B \rightarrow -0 \\ 00000000B \rightarrow +0 \end{cases}$

(2) 反码 最小 10000000B→-127

最大 01111111B→+127

两个零 $\begin{cases} 11111111B \rightarrow -0 \\ 00000000B \rightarrow +0 \end{cases}$

(3) 补码 最小 10000000B→-128

最大 01111111B→+127

一个零 00000000B→0

(二) 十六进制数表示的范围

1. 无符号数