



免费提供
电子教案

高等院校规划教材
计算机科学与技术系列

微机原理与接口技术

主 编 何珍祥
副主编 张明新



机械工业出版社
CHINA MACHINE PRESS



TP36/525

2009

高等院校规划教材·计算机科学与技术系列

微机原理与接口技术

主 编 何珍祥
副主编 张明新
参 编 董健康 安德智
 索国瑞 魏 芸

机械工业出版社

本书以 Intel 8086 为基础,介绍了微型计算机的结构、典型微处理器、存储器技术、8086 CPU 指令系统、汇编语言程序设计、输入/输出接口、微型计算机总线技术、人机交互设备及接口等知识,对微型计算机应用系统的设计和嵌入式系统也作了简单介绍。本书以微型计算机的关键技术(如微处理器、Cache、存储管理、中断、DMA 和系统总线等)作为重点,并结合实例进行了分析,书后的配套实验可使读者熟练掌握关键技术要点和应用方法,引导读者逐步掌握计算机硬件电路分析、应用和程序设计的能力。

本书内容丰富、应用性强,并提供配套的多媒体教学课件,可作为高等院校计算机及电类相关专业的教材或技术人员的培训教材,也可作为从事微型计算机科研、生产、教学和应用开发的科技人员的自学或参考用书。

图书在版编目(CIP)数据

微机原理与接口技术 / 何珍祥主编. —北京:机械工业出版社, 2009.4

(高等院校规划教材·计算机科学与技术系列)

ISBN 978-7-111-26823-9

I. 微… II. 何… 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(2009)第 054884 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:陈皓 常建丽

责任印制:杨曦

保定市中国画美凯印刷有限公司印刷

2009 年 5 月·第 1 版第 1 次印刷

184mm×260mm·23.25 印张·571 千字

0001—3000 册

标准书号:ISBN 978-7-111-26823-9

定价:39.00 元

凡购本书,如有缺页,倒页,脱页,由本社发行部调换

销售服务热线电话:(010) 68326294 68993821

购书热线电话:(010) 88379639 88379641 88379643

编辑热线电话:(010) 88379753 88379739

封面无防伪标均为盗版

出版说明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革，计算机学科的课程体系和教学内容趋于更加科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21世纪高等院校计算机教材系列”、“21世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级、省部级的精品教材。

为了进一步满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术以及计算机基础教育等系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列和信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方法讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。同时本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”建设。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师们能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前 言

随着计算机技术的飞速发展，微型计算机在工业、农业、国防、科研、教育、管理等领域得到了广泛应用。近年来，随着数字信号处理（DSP）和嵌入式系统的发展，已将微型计算机的应用推广到了生活电子产品中。

本书针对目前的应用实际，以 16 位的 8086 为主介绍了微型计算机原理与接口技术。本书详细介绍了 80x86/Pentium 的结构、存储器技术、8086 CPU 指令系统、汇编语言程序设计、输入/输出接口、微型计算机总线技术和人机交互设备及接口。本书有如下几个主要特点：

1) 突出实验能力。书中的每一部分都通过实例展开，在附录 A 提供了验证型、设计型和综合型实验。

2) 突出实际动手构建应用系统的能力。书中重点讲解了计算机硬件系统和 I/O 处理技术，体现以能力为本位的教学思想。例如，在存储器技术中介绍了存储器芯片的选配、存储器与 CPU 的连接，在 A/D 及 D/A 接口中介绍了 A/D 转换器的选择原则，在微型计算机应用系统中介绍了常用的 DSP 芯片、PCI 桥芯片、USB 芯片及连接成的实际系统。在硬件选择中坚持基本原则与最新元件的推荐相结合，最后通过实际应用系统案例来达到理论与实践的统一。

3) 突出接口编程能力的训练。在微型计算机应用系统中，中断编程、输入/输出接口的编程及在高级语言中对这些数据的处理能力是系统开发的软件基础。本书的指令系统部分详细介绍了相关指令，在汇编语言程序设计中增加了中断服务程序的设计与与 C 语言的连接，在输入/输出和常用可编程接口芯片中都提供了完整的应用程序，并用微型计算机应用系统中的驱动程序的编写来实现这一目标。

4) 突出内容的实用性和先进性。本书在大量介绍微型计算机系统实用内容的基础上还增加了一些新技术的介绍。在总线技术部分介绍了 PCI Express、SCSI 总线、USB 总线、IEEE1394 总线，在微型计算机应用系统中介绍了利用新元件构建的实际应用系统及嵌入式系统的基本知识。读者在学完本书之后，应具备分析和设计微型计算机应用系统的能力，能开展微型计算机应用系统的研发工作，为后续的专业学习和研究奠定基础。

本书可作为高等院校计算机及电类相关专业的教材或技术人员的培训教材，也可作为从事微型计算机科研、生产、教学和应用开发的科技人员的自学或参考用书。

本书由何珍祥统稿，第 1 章、附录 B 和附录 C 由张明新编写，第 2、10 章由安德智编写，第 3、9 章由董健康编写，第 4、5 章由索国瑞编写，第 6、11 章和附录 A 由何珍祥编写，第 7、8 章由魏芸编写。

由于作者水平有限，书中难免存在不妥之处，敬请读者批评指正。

编 者

目 录

出版说明

前言

第 1 章 微型计算机系统	1
1.1 微型计算机的发展	1
1.2 数据表示与数字信息	1
1.2.1 数据格式及机器数	1
1.2.2 数字信息编码的概念	5
1.3 微型计算机系统的基本组成	8
1.3.1 微型计算机的硬件结构	9
1.3.2 微型计算机的软件系统	11
1.4 小结	14
1.5 练习题	14
第 2 章 典型微处理器	15
2.1 8086 CPU 的内部结构	15
2.1.1 8086 CPU 的内部功能结构	15
2.1.2 8086 CPU 的寄存器结构	16
2.1.3 8086 总线的工作周期	19
2.1.4 存储器组织	19
2.2 8086 CPU 的引脚功能与工作模式	22
2.2.1 工作模式	22
2.2.2 8086 的引脚信号和功能	23
2.2.3 8086 CPU 的时序	28
2.3 80x86/Pentium 系列 CPU 技术发展	33
2.3.1 80x86 系列微处理器简介	33
2.3.2 Pentium 系列微处理器简介	35
2.4 小结	44
2.5 练习题	44
第 3 章 存储器技术	46
3.1 存储器概述	46
3.1.1 存储器的分类	46
3.1.2 存储器的主要性能参数	47
3.1.3 存储系统的层次结构	49
3.2 读写存储器	50
3.2.1 静态读写存储器 (SRAM)	50
3.2.2 动态读写存储器 (DRAM)	53

3.2.3	只读存储器 (ROM)	55
3.2.4	可擦编程只读存储器 (EPROM)	55
3.2.5	电可擦编程只读存储器 (EEPROM, E ² PROM)	56
3.2.6	闪速电可擦编程只读存储器 (闪速 FLASH)	57
3.3	存储器的连接	57
3.3.1	存储器的扩展	57
3.3.2	存储器寻址	59
3.3.3	存储器芯片的选配	60
3.3.4	存储器与 CPU 的连接	61
3.4	存储器管理	63
3.4.1	IBM PC/XT 中的存储空间分配	63
3.4.2	扩展存储器及其管理	64
3.5	内部存储器技术的发展	64
3.5.1	多模块交叉存储器	64
3.5.2	高速缓冲存储器 (Cache)	64
3.6	外部存储器	67
3.6.1	硬盘及硬盘驱动器	67
3.6.2	光盘存储器	70
3.7	小结	73
3.8	练习题	73
第 4 章	8086 CPU 指令系统	75
4.1	指令格式与寻址方式	75
4.1.1	指令格式	75
4.1.2	寻址方式	76
4.2	8086 CPU 指令系统	84
4.2.1	数据传送类指令	84
4.2.2	输入/输出类指令	89
4.2.3	算术运算类指令	90
4.2.4	位操作类指令	99
4.2.5	串操作类指令	104
4.2.6	控制转移类指令	107
4.2.7	处理器控制类指令	115
4.3	Pentium 系列 CPU 的指令系统	117
4.3.1	Pentium 系列 CPU 的新增寻址方式	117
4.3.2	Pentium 系列 CPU 的新增指令	118
4.4	小结	118
4.5	练习题	119
第 5 章	汇编语言程序设计	121
5.1	汇编语言概述	121

5.1.1	汇编语言源程序的格式	121
5.1.2	汇编语言的编程环境	123
5.2	汇编语言的伪指令	125
5.2.1	符号定义伪指令	125
5.2.2	数据定义伪指令	127
5.2.3	段定义伪指令	134
5.2.4	过程定义伪指令	137
5.3	汇编语言程序设计基础	138
5.3.1	程序设计的一般步骤	138
5.3.2	程序设计的基本方法	140
5.3.3	子程序设计与调用技术	150
5.3.4	DOS 功能子程序的调用	155
5.4	中断服务程序设计	157
5.5	模块化程序设计	162
5.5.1	模块化程序设计简介	162
5.5.2	多模块程序设计	164
5.5.3	汇编程序与 C 语言程序的连接	165
5.6	小结	170
5.7	练习题	170
第 6 章	输入/输出接口	173
6.1	输入/输出接口概述	173
6.1.1	输入/输出接口电路	173
6.1.2	CPU 与外设数据传送的方式	175
6.1.3	输入/输出端口的编址方式	177
6.2	中断系统	178
6.2.1	中断系统的基本概念	178
6.2.2	可编程中断控制芯片 8259A	182
6.2.3	8259A 的编程及应用	190
6.3	DMA 控制技术	194
6.3.1	可编程 DMA 控制芯片 8237A	194
6.3.2	8237A 的编程及应用	198
6.4	小结	204
6.5	练习题	205
第 7 章	常用可编程接口芯片	207
7.1	并行接口	207
7.1.1	并行通信与并行接口	207
7.1.2	可编程并行通信接口芯片 8255A	208
7.1.3	8255A 的编程及应用	220
7.2	串行接口	224

7.2.1	串行通信与串行接口	225
7.2.2	可编程串行通信接口芯片 8251A	228
7.2.3	8251A 的编程及应用	231
7.3	定时器/计数器	237
7.3.1	可编程定时器/计数器 8253A	237
7.3.2	8253A 的编程及应用	244
7.4	小结	247
7.5	练习题	247
第 8 章	A/D 及 D/A 接口	249
8.1	D/A 及 A/D 转换器概述	249
8.2	典型 D/A 转换器及其与 CPU 的接口	250
8.2.1	8 位 D/A 转换器 DAC 0832	251
8.2.2	DAC 0832 与 CPU 的接口	253
8.3	典型 A/D 转换器及其与 CPU 的接口	254
8.3.1	A/D 转换器 ADC 0809	256
8.3.2	A/D 转换器的选择原则	258
8.3.3	A/D 转换器与 CPU 的接口	258
8.4	小结	261
8.5	练习题	261
第 9 章	微型计算机总线技术	263
9.1	总线	263
9.1.1	微型计算机总线简介	263
9.1.2	总线分类和总线标准	264
9.1.3	微型计算机总线技术的现状和发展趋势	266
9.2	系统总线	267
9.2.1	ISA 总线	267
9.2.2	EISA 总线	268
9.2.3	PCI 总线	268
9.2.4	AGP 总线	271
9.2.5	新型总线 PCI Express	274
9.3	外总线	276
9.3.1	RS-232-C 总线	276
9.3.2	IEEE-488 总线	280
9.3.3	SCSI 总线	281
9.3.4	USB 总线	282
9.3.5	IEEE 1394 总线	286
9.4	小结	288
9.5	练习题	288
第 10 章	人机交互设备及接口	290
10.1	显示接口	290

10.1.1	CRT 显示器及其接口	290
10.1.2	LCD 显示器及其接口	294
10.1.3	LED 显示器及其接口	295
10.2	键盘、鼠标接口	297
10.2.1	键盘接口	297
10.2.2	鼠标接口	300
10.3	打印机及其接口	301
10.3.1	常用打印机及其工作原理	301
10.3.2	主机与打印机接口	305
10.3.3	打印机编程应用	306
10.4	其他外设	307
10.4.1	扫描仪	307
10.4.2	数码相机	308
10.4.3	光盘刻录机	309
10.4.4	外围设备的发展方向	311
10.5	小结	312
10.6	练习题	312
第 11 章	微型计算机应用系统	313
11.1	微型计算机应用系统的设计	313
11.1.1	简介	313
11.1.2	微型计算机应用系统设计举例	316
11.2	PCI 总线和 USB 总线接口设计	317
11.2.1	PCI 总线与 DSP 通信接口设计	317
11.2.2	USB 总线与 DSP 通信接口设计	320
11.3	Windows 驱动程序设计	323
11.3.1	驱动程序概述	323
11.3.2	PCI 设备 VxD 驱动程序设计	325
11.3.3	USB 设备 WDM 驱动程序设计	327
11.4	嵌入式系统	330
11.4.1	简介	330
11.4.2	嵌入式系统硬件的组成	332
11.4.3	嵌入式系统软件的开发	333
11.5	小结	334
11.6	练习题	334
附录	336
附录 A	实验	336
A.1	实验一 调试工具 Debug 的使用	336
A.2	实验二 顺序结构程序设计及上机过程	337
A.3	实验三 循环与分支结构程序设计	338

A.4	实验四	子程序结构程序设计	339
A.5	实验五	系统调用	339
A.6	实验六	可编程中断控制器 8259A 实验	340
A.7	实验七	可编程 DMA 控制器 8237A 实验	341
A.8	实验八	可编程并行通信接口芯片 8255A 实验	342
A.9	实验九	可编程串行通信接口芯片 8251A 实验	344
A.10	实验十	可编程定时器/计数器 8253A 实验 (一)	345
A.11	实验十一	可编程定时器/计数器 8253A 实验 (二)	346
A.12	实验十二	数/模 (D/A) 转换器 DAC 0832 实验	348
A.13	实验十三	小直流电动机控制实验	349
A.14	实验十四	步进电动机控制实验	350
A.15	实验十五	温度测量实验	350
	附录 B	DEBUG 工具	352
	附录 C	DOS 系统功能调用 (INT 21H)	355
	参考文献	359

第 1 章 微型计算机系统

1.1 微型计算机的发展

计算机是一种自动、高速、精确地进行信息处理的现代化电子设备。从 1946 年第一台计算机诞生至今仅 50 多年，计算机已经由电子管时代、晶体管时代、中小规模集成电路时代，发展到大规模、超大规模集成电路时代，现正在向第五代计算机发展。从 70 年代初，由大规模集成电路组成的微型计算机问世，到现在仅 30 年左右，微处理器已经推出了四代产品：4 位微处理器、8 位微处理器、16 位微处理器和 32 位以上微处理器。在微型计算机的发展过程中，最成功也最具有影响力的是 IBM PC 系列微机，又称 PC (Personal Computer)。1982 年，IBM 公司推出了以 Intel 8086 CPU 为处理器的 IBM PC。1983 年，IBM 公司又推出了采用 Intel 8088 CPU 为处理器的 IBM PC/XT。这两种机型的内存均为 1MB，支持单任务的操作系统。1984 年，以 Intel 80286 为 CPU 的 16 位增强型 PC IBM PC/AT 上市，其内存可达到 8MB，并支持多任务多用户操作系统。继 IBM PC/AT 之后，Intel 公司推出了 32 位微处理器 80386 和 80486，由 80386 和 80486 CPU 构成的 PC 80386 和 PC 80486 内存物理地址空间可达 4GB，支持多任务多用户操作系统，并增加了高速缓冲存储器 Cache。80486 以后的 CPU，名称上改为奔腾 (Pentium) 系列，其字长已达到 64 位，运算速度和功能、性能与 PC 80486 机相比有很大提高。

在微机家族中，单片微机的发展同样引人注目。单片机是把 CPU、一定容量的存储器和 I/O 接口电路集成到一片芯片上，构成具有计算机完整功能的一种微机。单片机的字长已由 4 位、8 位发展到目前的 16 位，存储器可以扩充到 64KB，同时还含有模数控制器 (ADC) 和数模控制器 (DAC)，并且有功能很强的指令系统。单片机在工业控制和智能仪表中得到了广泛的应用。

1.2 数据表示与数字信息编码

计算机中的数据分为数值型数据和非数值型数据。数值型数据用于表示数值的正负、大小，可进行数学运算。非数值型数据主要用于表示文字、符号和图形等。在计算机里，无论是数值型数据还是非数值型数据，都是以二进制代码的形式存储、传输和处理，它们与电子元器件的两种状态相对应。但由于二进制数在表示较大数据时显得太冗长，人们阅读时不方便，因此在阅读和编程时，往往借用八进制或十六进制数来表示机器码或机器数。

1.2.1 数据格式及机器数

1. 真值表示和机器数表示

在日常的书写习惯中，往往用正负符号加绝对值表示数值，这种形式表示的数值在计算

机技术中被称为真值。由于计算机采用数字化信息表示方法，对正负号也需要数字化。这种在计算机中使用的、连同符号一起数字化了的数，就称为机器数。例如：

- + 58 用十进制表示的真值；
- +111010 用二进制表示的真值；
- 0111010 用二进制表示机器数。

机器数的位数与计算机的字长有关。为了方便讨论，在后面的例子中，取计算机的字长为 8 位。要利用计算机对数值型数据进行运算，必须对其进行编码。常用的数值型数据的编码方法有：原码、补码、反码以及 BCD 码。

(1) 原码表示法

原码表示是一种比较直观的机器数表示方法。约定数码序列中的最高位为符号位，符号位为 0 时表示该数为正，为 1 时表示该数为负，其余有效值部分则用二进制的绝对值表示。

若定点小数的原码序列为 $[X]_{\text{原}} = X_f X_1 \dots X_n$ ，则

$$[X]_{\text{原}} = \begin{cases} X & 0 \leq X < 1 \\ 1 - X = 1 + |X| & -1 < X \leq 0 \end{cases}$$

若定点整数的原码序列为 $[X]_{\text{原}} = X_f X_n \dots X_0$ ，则

$$[X]_{\text{原}} = \begin{cases} X & 0 \leq X < 2^n \\ 2^n - X = 2^n + |X| & -2^n < X \leq 0 \end{cases}$$

分析这两个定义式可以得出以下结论：

1) 真值零在原码表示中可以有两种形式，即正零和负零。

小数： $[+0]_{\text{原}} = 0.00\dots 0$ ； $[-0]_{\text{原}} = 1.00\dots 0$

整数： $[+0]_{\text{原}} = 000\dots 0$ ； $[-0]_{\text{原}} = 100\dots 0$

正零和负零的真值含义相同，这只是原码的定义造成的结果。注意，这里的小数点是默认的，机器数是没有小数点的，以下出现的小数点只是为读者阅读而写的。

2) 符号位不是数值的一部分，它们是人为约定“0 正 1 负”，所以符号位在运算过程中需要单独处理，不能当作数值的一部分直接参与运算。

3) 对于小数 $-1 < X < 1$ ，即表示范围限制在 $|X| < 1$ 之内。对于整数， $-2^n < X < 2^n$ ，即表示范围限制在 $|X| < 2^n$ 之内。

原码表示中用绝对值表示数值，比较直观。但对于应用得最多的加减运算，则不太方便。

(2) 补码表示法

其定义如下：

1) 若定点小数的补码序列为 $[X]_{\text{补}} = X_f X_1 \dots X_n$ ，则

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

式中，X 为真值， $[X]_{\text{补}}$ 是采取补码表示的机器数。

例如：若 $X = 0.1010$ ，则 $[X]_{\text{补}} = 0.1010$

例如：若 $X = -0.1010$ ，则 $[X]_{\text{补}} = 2 - 0.1010 = 1.0110$

2) 若定点整数的补码序列为 $[X]_{\text{补}} = X_f X_n \dots X_0$ 则

$$[X]_{\text{补}} = \begin{cases} X & 0 \leq X < 2^n \\ 2^{n+1} + X = 2^{n+1} - |X| & -2^n < X \leq 0 \end{cases} \quad (\text{mod } 2^{n+1})$$

手工求补码的方法：如果真值为正，则正数的补码与原码表示形式相同；如果真值为负，则将负号用“1”表示在首位，原码的数值部分逐位取反，并且在末位加1，写在符号位之后即得到负数的补码。已知补码求真值的方法是：如果补码的符号位是“0”，说明该补码是正数，则它的真值就是补码本身；如果补码的符号位为“1”，说明该补码是负数。因此先记一个“-”号，然后把补码逐位取反，并且在末位加1，写在符号“-”之后即得到该补码的真值。

3) 补码的性质

- 在补码表示中，最高位 X_f (符号位) 表示数的正负，在形式上与原码相同，即0正1负。但补码的符号位是数值的一部分。
- 在补码表示中，数0只有一种表示，即00...。读者可以自己证明。
- 负数补码的表示范围比原码稍宽，多一种数码组合100...0。
- 负数补码表示的实质是将负数映射到正数域，因而可实现化减为加的目的。

(3) 反码表示法

1) 若定点小数的反码形式为 $[X]_{\text{反}} = X_f X_1 \dots X_n$ ，则

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

2) 若定点整数的反码形式为 $[X]_{\text{反}} = X_f X_{n-1} \dots X_0$ ，则

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

真值的零在反码中也有正零和负零两种形式：

小数： $[+0]_{\text{反}} = 0.00\dots0$ ； $[-0]_{\text{反}} = 1.11\dots1$

整数： $[+0]_{\text{反}} = 000\dots0$ ； $[-0]_{\text{反}} = 111\dots1$

反码的表示范围与原码相同。

手工求反码的方法：正数的反码与它的原码相同，负数的反码符号位填“1”，将数值位逐位求反后写在符号位之后就完成了。

(4) 机器数的位数格式

在工程上，由于计算机的字长关系，因此真值的机器数表示就与字长有关系。例如8位机，它的一个字长的机器数就是8位，两个字长就是16位等。如果真值的数值位加符号位不足8位，必须按照8位表示，超过8位不足16位必须按两个字长表示为16位，以下类推。因此，真值在表示为机器数的过程中需要首先化为8位、16位或者更多个机器字长，然后再转换为机器数。

2. 定点表示与浮点表示

实际的真值可能既有整数部分又有小数部分，在进行加减运算时必须先将小数点位置对准方可进行。这样就提出一个如何确定小数点位置的问题。根据小数点位置是否固定，机器数的格式可分为定点表示与浮点表示两类。

(1) 定点表示法

定点表示法约定小数点位置固定不变。为了处理方便，一般只采取三种简单的约定，相应有以下三种类型的定点数表示法。

1) 无符号整数。无符号整数是把符号位略去的正整数，因此编码的所有数位都用来表示数值的大小，小数点在最低位之后，整个数码是一个整数。

二进制的无符号整数 $X_n X_{n-1} \cdots X_0$ ，其表示范围是 $0 \sim (2^{n+1}-1)$ ，分辨率为 1，后者表明了它的绝对精度。

2) 带符号定点整数。代码序列 $X_f X_{n-1} \cdots X_0$ ， X_f 为符号位，小数点位置在最低位 X_0 之后。

原码定点整数表示范围： $-(2^n - 1) \sim 2^n - 1$

补码定点整数表示范围： $-2^n \sim 2^n - 1$

分辨率（非零最小正数）：1

3) 带符号定点小数。代码序列 $X_f X_1 \cdots X_n$ ， X_f 为符号位，小数点位置默认在符号位 X_f 之后，即此数为纯小数。在原码中， $X_1 \cdots X_n$ 是数值部分，常称之为尾数，并将数值部分的最高位 X_1 ，称为最高数值位或最高有效位（注意：实际的机器数是没有小数点的）。

原码定点小数表示范围： $-(1 - 2^{-n}) \sim (1 - 2^{-n})$

补码定点小数表示范围： $-1 \sim (1 - 2^{-n})$

分辨率： 2^{-n}

(2) 浮点表示法

定点数表示法比较简单，实现定点运算的硬件成本较低，广泛用于低档机中。但是，在有限位数的定点数中，表示范围与精度两项指标是有矛盾的。例如，一个 16 位无符号正整数，表示范围为 $0 \sim 65535$ ，分辨率为 1。如果要求它表示的数的范围扩大 10 倍，则分辨率将升至 10，即精度降低 1/10。在同一个定点运算程序中，比例因子还要适用于多个数据，因而常常不能兼顾两方面的要求。

在工程实际中，测量的精确度常常由多少位有效数字表示，它们的单位可能相差悬殊。例如，星球之间的距离很远，分辨率以光年为单位；而半导体制造工艺中的加工精度以微米计，甚至是纳米，每个芯片所占的图形尺寸不到 1cm^2 。这就要求把比例因子以恰当的形式包含在数码中，在有限位数的前提下，最大地表示出数据的范围。这样做的实质是让小数点的位置不固定，使之根据需要而浮动。这种表示法称为浮点表示法。

1) 浮点数格式。典型的浮点数格式如图 1-1 所示，浮点数的代码分为两个组成部分：阶码 E 与尾数 M，浮点数的符号由尾数中的符号位 M_f 表示。浮点数的真值为

$$N = \pm R^E \cdot M$$

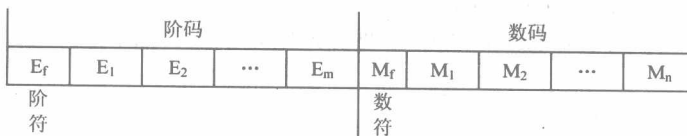


图 1-1 典型的浮点数格式

其中， R 是阶码的底，一般与尾数的基数相同。对于同一种系统， R 是隐含约定的常数，并不直接表示在浮点代码之中。在实际的计算机中， R 多取为 2、4、8、16 等，其中以

$R = 16$ 最为普遍。但在教科书中为了教学的方便常约定尾数为二进制，相应地取 R 为 2。 E 是阶码，为二进制正负整数，可用补码或原码表示。 R^E 相当于尾数的比例因子，若 E 为正，表明尾数 M 将被扩大。若 E 为负，表明 M 将被缩小。 M 是尾数，为正负定点小数，小数点默认在符号位 M_f 后，尾数可用补码或原码表示。

2) 尾数的规格化问题。一个数的浮点数表示可以有多种写法。例如， $2^{010} \times 0.0101100$ 可以表示为 $2^{001} \times 0.1011000$ ， $2^{011} \times 0.0010110$ ，…。为了充分利用尾数部分的有效位数，使精度进一步提高，尾数要采用规格化表示。以 $R = 2$ 为例，则规格化尾数需满足条件： $1/2 \leq |M| \leq 1$ 。上面的 $2^{001} \times 0.1011000$ 就是规格化的写法，它的尾数的补码就是 01011000。其中第一位的“0”表示数符，第二位的“1”表示尾数的最高数值位，小数点默认在符号位之后。

综上可知，浮点数实际上由两个定点数组成，阶码是一个定点整数，尾数是一个定点小数。它相当于将数规范为定点小数 M 乘以比例因子 R^E 。

3) 浮点数表示范围与精度。设阶码部分和阶符共 $m+1$ 位，用补码表示，底为 2；尾数共 $n+1$ 位，含一位数符，用补码表示，尾数进行了规格化。

浮点数表示范围： $-(2^m-1) \sim 2^m-1$

分辨率： 2^{-m}

浮点数的表示范围取决于阶码的位数，相对精度取决于尾数的位数。对比一下同样字长的定点数和浮点数，可以看出：浮点数能表示的数的范围要大得多，并且表示较小数值时同样有很高的分辨率。在实际使用中，常用 32 位短浮点数或 64 位、128 位浮点数进行高精度运算。计算机中浮点数的实际格式如图 1-2 所示。

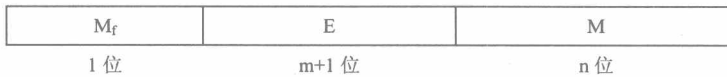


图 1-2 计算机中浮点数的实际格式

按照 IEEE 规定的浮点数标准，计算机浮点数的阶码和尾数的位数关系见表 1-1。

表 1-1 计算机浮点数的阶码和尾数的位数关系

	总 位 数	符 号 位	阶 码	尾 数
短浮点数	32	1	8	23
长浮点数	64	1	11	52
临时浮点数	80	1	15	64

1.2.2 数字信息编码的概念

1. 字符的表示

字符是非数值型信息的一种，也可用它来间接表示数值型数据。例如，可用字符和字符串构成程序设计语言来编写程序，其中包含用字符描述的数据。在程序被编译成可执行机器代码之前，往往要保持字符形式。又如，以字符为基础表示文字，可以描述范围广泛的绝大多数信息。

国际上广泛采用美国信息交换标准码 (American Standard Code For Information Interchange, ASCII)。ASCII 字符编码见表 1-2。

表 1-2 ASCII 字符编码

十六进制 制编码	字 符	十六进制 编 码	字 符	十六进 制编码	字 符	十六进 制编码	字 符	十六进 制编码	字 符
00	NUL	1A	SUB	34	4	4E	N	68	h
01	SOH	1B	ESC	35	5	4F	O	69	i
02	STX	1C	FS	36	6	50	P	6A	j
03	ETX	1D	GS	37	7	51	Q	6B	k
04	EOT	1E	RS	38	8	52	R	6C	l
05	ENQ	1F	US	39	9	53	S	6D	m
06	ACK	20	SPACE	3A	:	54	T	6E	n
07	BEL	21	!	3B	;	55	U	6F	o
08	BS	22	“	3C	<	56	V	70	p
09	HT	23	#	3D	=	57	W	71	q
0A	LF	24	\$	3E	>	58	X	72	r
0B	VT	25	%	3F	?	59	Y	73	s
0C	FF	26	&	40	@	5A	Z	74	t
0D	CR	27	'	41	A	5B	[75	u
0E	SO	28	(42	B	5C	\	76	v
0F	SI	29)	43	C	5D]	77	w
10	DLE	2A	*	44	D	5E	^	78	x
11	DC1	2B	+	45	E	5F	_	79	y
12	DC2	2C	,	46	F	60	`	7A	z
13	DC3	2D	-	47	G	61	a	7B	{
14	DC4	2E	.	48	H	62	b	7C	
15	NAK	2F	/	49	I	63	c	7D	}
16	SYN	30	0	4A	J	64	d	7E	~
17	ETB	31	1	4B	K	65	e	7F	DEL
18	CAN	32	2	4C	L	66	f		
19	EM	33	3	4D	M	67	g		

注: NUL—空白 SOH—序始 STX—文始 ETX—文终 EOT—送毕 ENQ—询问
 ACK—承认 BEL—警铃 BS—退格 TAB—横表 LF—换行 VT—纵表 FF—换页
 CR—回车 SO—移出 SI—移入 DLE—转义 DC1—机控 1 DC2—机控 2
 DC3—机控 3 DC4—机控 4 NAK—否认 SYN—同步 ETB—组终 CAN—作废
 EM—载终 SUB—取代 ESC—扩展 FS—卷隙 GS—群隙 RS—录隙
 US—元隙 SPACE—空格 ~—ESC 换码 DEL—删除

ASCII 字符集共有 128 种常用字符, 其中有数字字符 0~9, 大小写英文字母, 一些在算式中 and 文本中常用的符号 (如四则运算符、括号、标点符号、标识符等), 还有一些控制字符, 这些字符种类大致满足了西文文字、编程语言、常见控制命令等的需要。每个 ASCII 码字符用 7 位编码 ($D_6 \sim D_0$), 高位 D_7 往往默认为 0 或作为奇偶校验位。因此存储器以字节作