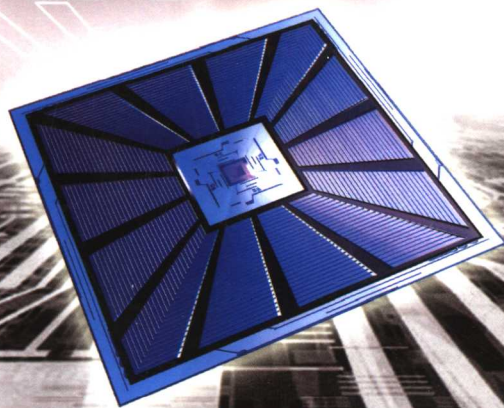


21世纪重点大学规划教材

余春暄 编著

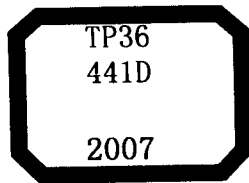
80x86/Pentium 微机原理及接口技术



机械工业出版社
CHINA MACHINE PRESS



附赠光盘



21 世纪重点大学规划教材

80x86/Pentium 微机原理 及接口技术

余春暄 编著

机械工业出版社

本书主要介绍从 8086 到 Pentium 4 微处理器的结构和特点、寻址方式、指令系统及汇编语言程序设计, 以及微型计算机各组成部分的原理、常用接口技术及其应用。

本书以培养学生应用能力为主要目标, 强调基本知识和基本技术, 以及分析问题和解决问题的方法, 在传统内容的基础上力求反映微型计算机及微处理器发展的新技术, 立足于通俗易懂、由浅入深、举一反三, 既可作为高等院校计算机及相关专业的计算机原理教材, 又可供计算机硬件或软件开发人员参考。

为了配合教师课堂教学和学生课后学习, 本书配备了用 Authorware 等软件制作的多媒体辅助教学课件, 利用动画帮助读者理解。课件中还配有教学大纲、自测练习, 使读者明确整个学习内容, 了解学习效果。

图书在版编目 (CIP) 数据

80x86/Pentium 微机原理及接口技术/余春暄编著. —北京: 机械工业出版社, 2007.1

(21 世纪重点大学规划教材)

ISBN 978-7-111-20743-6

I. 8… II. 余… III. 微处理器—高等学校—教材 IV. TP332

中国版本图书馆 CIP 数据核字 (2007) 第 005200 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 车 忱

责任印制: 李 妍

北京铭成印刷有限公司印刷

2007 年 2 月第 1 版·第 1 次印刷

184mm×260mm·22.75 印张·562 千字

0001—5000 册

标准书号: ISBN 978-7-111-20743-6

ISBN 978-7-89482-108-9 (光盘)

定价: 37.00 元(含 1CD)

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

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

编辑热线电话: (010) 88379739

封面无防伪标均为盗版

出版说明

“211 工程”是“重点大学和重点学科建设项目”的简称，是国家“九五”期间惟一的教育重点项目。

进入“211 工程”的 100 所学校拥有全国 32% 的在校本科生、69% 的硕士生、84% 的博士生，以及 87% 的有博士学位的教师；覆盖了全国 96% 的国家重点实验室和 85% 的国家重点学科。相对而言，这批学校中的教授、教师有着深厚的专业知识和丰富的教学经验，其中不少教师对我国高等院校的教材建设做过很多重要的工作。为了有效地利用“211 工程”这一丰富资源，实现以重点建设推动整体发展的战略构想，机械工业出版社推出了“21 世纪重点大学规划教材”。

本套教材以重点大学、重点学科的精品教材建设为主要任务，组织知名教授、教师进行编写。教材适用于高等院校计算机及其相关专业，选题涉及公共基础课、硬件、软件、网络技术 etc，内容紧密贴合高等院校相关学科的课程设置和培养目标，注重教材的科学性、实用性、通用性，在同类教材中具有一定的先进性和权威性。

为了体现建设“立体化”精品教材的宗旨，本套教材为主干课程配备了电子教案、学习指导、习题解答、课程设计、毕业设计指导等内容。

机械工业出版社

前 言

电子技术和计算机技术的迅猛发展, 引发了人类生活、学习和科学研究各个领域的技术革命。现代社会和人类生活越来越离不开计算机, 计算机知识和应用技能已成为人类知识的重要组成部分, 也成为各大专院校大部分电类、机电类、生物工程等专业的必修课程。

计算机原理与应用课程知识点多, 初学者常感到课程难学、作业难做。本教材是在参阅了当前国内外有关微型计算机的大量资料基础上, 根据作者多年的教学实践和科学研究的经验编写的, 立足于通俗易懂、由浅入深、举一反三, 使其既可用作高等院校计算机及相关专业“计算机原理及应用”课程的教材, 又可供计算机硬件或软件开发人员参考。本教材的主要特点有:

1) 结构清楚、重点突出、循序渐进、实例丰富。

2) 以目前最为普及的 Intel 80x86/Pentium 系列计算机系统作为背景, 详细介绍了计算机的组成结构、工作原理、指令系统、接口技术等, 为后续课程及计算机应用、开发打下良好的基础。

3) 为了配合教师课堂教学和学生课后学习, 本教材配备了用 Authorware 等软件制作的多媒体辅助教学课件(在随书光盘中), 利用动画帮助读者理解。课件中还配有教学大纲、自测练习, 使读者明确整个学习内容, 了解学习效果。

4) 本教材注重基础, 强调理论和实践相结合。重点介绍 8086/8088 处理器及外围接口技术的原理和应用方法, 适当介绍了计算机技术的发展历程, 从而使读者加深对计算机系统工作过程的理解, 掌握用计算机解决实际问题的方法。

本教材共分为 8 章。第 1~4 章介绍计算机的组成、微处理器的结构、计算机中数和编码的表示方法, 帮助读者建立计算机系统的整体概念, 通过处理器执行程序过程的例子, 了解微处理器的工作过程; 以 8086/8088 微处理器为切入点, 重点介绍 8086/8088 微处理器的内部结构、内部寄存器、工作模式、引脚定义、存储器组织和系统组成, 在此基础上介绍了 80x86 及 Pentium 系列微处理器的发展和特点, 以及 80x86 / Pentium 系列处理器的指令格式、寻址方式、指令系统、汇编语言程序设计以及调试手段, 最后给出了汇编语言与 C/C++ 混合编程的方法。第 5 章介绍计算机中常用的半导体存储器的分类与接口设计方法。第 6~8 章介绍计算机接口技术, 对计算机接口概念进行了详细阐述, 通过应用实例介绍简单接口技术和可编程接口技术。

本教材采用的逻辑符号部分是国外流行符号, 其与国家标准的对照请参阅附录 D。

本教材在编写过程中参考了大量文献, 在参考文献中已尽量列出, 但仍有部分资料因原始出处不详未能列出, 在此向这些文献的作者表示感谢。在编写过程中得到了左国玉、施远征、韦燕凤等老师的帮助。李展鹏、李峰、彭靖漩、杨洋等同学也做了大量的工作, 在此一并表示衷心感谢。由于笔者水平有限, 书中难免有错误和不妥之处, 请读者批评指正。

编 者

目 录

出版说明

前言

第 1 章 计算机基础	1
1.1 概述	1
1.1.1 微计算机的发展	1
1.1.2 常用术语	2
1.1.3 正文中使用的符号	3
1.2 计算机中数与编码的表示方法	3
1.2.1 进制表示	3
1.2.2 定点和浮点表示	5
1.2.3 有符号数的表示	6
1.2.4 编码的表示	8
1.3 微型计算机结构及工作原理	9
1.3.1 微处理器	9
1.3.2 微型计算机	12
1.3.3 微型计算机系统	13
1.3.4 计算机中的指令执行过程	13
1.4 微机系统中采用的先进技术	15
1.4.1 流水线技术	15
1.4.2 高速缓冲存储技术	16
1.4.3 CISC 和 RISC	17
1.4.4 多核心技术	18
1.5 习题与思考	18
第 2 章 80x86 及 Pentium 微处理器	20
2.1 Intel 系列微处理器概述	20
2.2 8086/8088 微处理器	22
2.2.1 8086/8088 编程结构	22
2.2.2 8086/8088 的工作模式与引脚定义	28
2.2.3 8086/8088 的时序与总线周期	32
2.2.4 8086/8088 的存储器组织	36
2.2.5 8086/8088 的堆栈组织	39
2.2.6 8086/8088 的系统组织	40
2.3 80286 到 Pentium 系列微处理器	44
2.3.1 概述	44
2.3.2 80x86 微处理器	50

2.3.3 Pentium 系列微处理器	54
2.3.4 双核微处理器	57
2.4 习题与思考	59
第 3 章 80x86 及 Pentium 指令系统	61
3.1 指令的结构	61
3.1.1 指令提供的信息	61
3.1.2 指令格式	62
3.2 寻址方式	64
3.2.1 寻址方式的定义	64
3.2.2 寻址方式分类	64
3.2.3 数据型操作数寻址方式	65
3.2.4 目标地址寻址方式	71
3.3 8086/8088 指令系统	73
3.3.1 传送类指令	74
3.3.2 算术运算类指令	81
3.3.3 逻辑运算类指令	92
3.3.4 位移类指令	94
3.3.5 转移类指令	97
3.3.6 串操作类指令	109
3.3.7 处理器类指令	113
3.4 80286 到 Pentium 增加指令介绍	114
3.4.1 80286 的增强与增加指令	115
3.4.2 80386/80486 的增强与增加指令	117
3.4.3 Pentium 系列处理器的增加指令	118
3.5 习题与思考	122
第 4 章 汇编语言程序设计	125
4.1 概述	125
4.1.1 计算机语言的分类	125
4.1.2 MASM 汇编语言	126
4.2 伪指令	128
4.2.1 汇编语言中数、符号、表达式的描述规范	129
4.2.2 处理器定义伪指令	132
4.2.3 模式定义伪指令	133
4.2.4 段定义伪指令	133
4.2.5 数据定义伪指令	138
4.2.6 符号定义伪指令	139
4.2.7 类型定义伪指令	140
4.2.8 过程定义伪指令	141
4.2.9 程序计数器与定位伪指令	142

4.2.10	条件汇编伪指令	143
4.2.11	记录与结构伪指令	143
4.2.12	模块定义伪指令	148
4.3	宏指令	149
4.3.1	宏指令定义	149
4.3.2	宏指令的应用	150
4.3.3	宏指令与子程序的区别	151
4.4	BIOS 和 DOS 的功能调用	151
4.4.1	BIOS 调用	152
4.4.2	DOS 软中断	152
4.4.3	DOS 系统功能调用	152
4.5	汇编语言程序设计与调试	154
4.5.1	汇编语言程序设计步骤	154
4.5.2	MASM 汇编语言的调试方法	155
4.6	汇编语言程序设计基本方法	156
4.6.1	顺序结构程序	156
4.6.2	分支结构程序	159
4.6.3	循环结构程序	164
4.6.4	子程序调用结构程序	169
4.6.5	综合应用程序设计举例	176
4.6.6	80x86 应用程序设计举例	177
4.7	汇编语言与 C/C++ 语言的混合编程	179
4.7.1	内嵌模块方法	179
4.7.2	外调模块方法	180
4.8	习题与思考	183
第 5 章	半导体存储器及其接口技术	187
5.1	计算机存储器概述	187
5.1.1	微型计算机中存储器的分类	187
5.1.2	半导体存储器的分类	188
5.1.3	半导体存储器的性能指标	190
5.1.4	计算机系统中常见的半导体存储器	190
5.1.5	存储卡技术介绍	192
5.2	半导体存储器结构与工作原理	195
5.2.1	半导体存储器的基本组成	195
5.2.2	随机存储器 RAM	196
5.2.3	只读存储器 ROM	203
5.2.4	非易失读/写存储器	205
5.3	半导体存储器接口设计	210
5.3.1	存储芯片的选择	211

5.3.2	存储器的地址分配	212
5.3.3	存储器的地址译码	213
5.3.4	存储器与 CPU 的信号连接	215
5.3.5	存储器接口设计举例	216
5.4	x86 存储器技术	221
5.4.1	虚拟存储器简介	221
5.4.2	80x86 中的 ROM 重复和影子 RAM	222
5.5	习题与思考	222
第 6 章	微型计算机接口技术	225
6.1	微型计算机接口结构与功能	225
6.1.1	接口的基本结构	226
6.1.2	接口的功能	226
6.1.3	x86 PC 系统中 I/O 地址映射	228
6.2	微处理器与外设数据传输控制方式	229
6.2.1	直接程序传输	229
6.2.2	查询程序传输	229
6.2.3	中断传输	230
6.2.4	DMA 传输	231
6.3	微计算机的中断系统	232
6.3.1	中断控制方式的优点	232
6.3.2	与中断有关的术语	233
6.3.3	中断过程	233
6.3.4	80x86 中断系统	238
6.4	微型计算机功能扩展总线和接口标准	239
6.4.1	一些常用的总线术语	240
6.4.2	总线的分类	240
6.4.3	80x86 系列微机中常用的总线和接口标准	241
6.5	习题与思考	248
第 7 章	简单接口电路设计	249
7.1	概述	249
7.1.1	数据锁存器	249
7.1.2	数据缓冲器	250
7.2	开关量输出接口设计	250
7.2.1	单个开关量输出接口	250
7.2.2	多个开关量输出接口	251
7.2.3	数码显示接口	252
7.3	开关量输入接口设计	255
7.3.1	单个开关量输入接口	255
7.3.2	多个开关量输入接口	255

7.3.3	键盘接口	255
7.4	D/A 转换接口	259
7.4.1	D/A 转换原理	259
7.4.2	D/A 转换器技术参数	261
7.4.3	DAC 0832 介绍	261
7.5	A/D 转换接口	264
7.5.1	A/D 转换原理	264
7.5.2	A/D 转换器的主要技术指标	267
7.5.3	ADC 0809 介绍	268
7.6	A/D 和 D/A 转换接口应注意的问题	271
7.7	习题与思考	272
第 8 章	可编程接口技术	274
8.1	可编程计数器 8253/8254	274
8.1.1	8253/8254 外部特点与功能	274
8.1.2	8253/8254 内部结构与工作原理	275
8.1.3	8253/8254 的控制字与初始化编程	276
8.1.4	8253/8254 的工作方式	278
8.1.5	8253/8254 应用举例	280
8.2	并行通信接口 8255A	284
8.2.1	8255A 引脚与内部结构	284
8.2.2	8255A 的工作方式及其初始化设置	286
8.2.3	8255A 各工作方式的功能特点说明	287
8.2.4	8255A 应用举例	291
8.3	串行通信接口 8250/8251	298
8.3.1	串行通信接口技术的概念	298
8.3.2	可编程异步通信接口 8250/16450	304
8.3.3	可编程串行通信接口芯片 Intel 8251A	314
8.4	可编程中断控制器 8259A	322
8.4.1	8259A 概述	322
8.4.2	8259A 的引脚特性	322
8.4.3	8259A 内部结构	323
8.4.4	8259A 的初始化编程	325
8.4.5	8259A 的工作编程	328
8.4.6	8259A 应用举例	330
8.5	DMA 控制器 8237A	336
8.5.1	8237A 的功能及引脚特性	336
8.5.2	8237A 内部寄存器及读写操作	338
8.5.3	8237A 初始化编程	341
8.5.4	8237A 应用举例	341

8.6 习题与思考	343
附录	348
附录 A 7 位 ASCII 码编码表	348
附录 B DEBUG 的常用命令	349
附录 C 多媒体辅助教学软件说明	350
附录 D 逻辑符号对照表	351
参考文献	352

第 1 章 计算机基础

计算机技术是 20 世纪最杰出的科技成果之一，它极大地改变了人类社会的生活、学习和工作的方式。本章介绍计算机的发展背景、体系结构、常用术语以及计算机中数和编码的表示方法，使读者了解计算机系统的整体概念和计算机的基本工作过程，为后续学习打下基础。

1.1 概述

1.1.1 微计算机的发展

“计算”是人类生活中最重要的活动之一。随着人类社会的发展与进步，计算量越来越大，而且越来越复杂，促使人类不断推出各种各样的计算工具，如我国唐宋时期出现了算盘，后来科学家又发明了计算尺、机械式计算器……随着电子技术的发展，在 1946 年，世界上第一台电子数字积分式计算机 ENIAC (Electronic Numerical Integrator And Calculator) 诞生了。著名数学家冯·诺伊曼 (Von Neumann) 在 ENIAC 计算机的设计制造期间，首次提出了“存储程序”的概念，从那时开始，这个概念一直延用至今。因此，人们把按照这一概念制造的计算机称作冯·诺伊曼计算机 (Von Neumann Machine)。冯·诺伊曼计算机的核心就是：

- (1) 指令和数据用二进制数表示。
- (2) 程序预存在存储器中，在执行时会将指令自动地逐条取出并分析执行。
- (3) 计算机的硬件由运算器、控制器、存储器、输入设备和输出设备五部分组成。

计算机的发展可分为：

第一代 (1945~1958 年) 真空管计算机。真空管体积大，并且非常耗电，例如 ENIAC 的功率为 130 000W。

第二代 (1958~1964 年) 晶体管计算机。

第三代 (1964~1971 年) 集成电路计算机。

第四代 (1971~今) 超大规模集成电路计算机。

目前正在向第五代人工智能计算机方向突破，其主要目标是希望实现更高程度上模拟人脑的思维功能。

现在，人们广泛使用的计算机是第四代计算机，它的发展是以微处理器的发展为基础的。微处理器的更新速度很快，几乎每两年集成度翻一番，每 2~4 年更新换代一次。可以看到：

1971~1973 年，第一代微处理器主要为 4 位或低档 8 位微处理器，其指令系统比较简单，运算能力差、速度慢。

1974~1978 年，第二代中高档 8 位微处理器，它比第一代微处理器有了较多的改进，集成度提高 1~4 倍，运算速度提高 10~15 倍，指令系统相对比较完善，已具有典型的计算

机体系结构以及中断、存储器直接读取（DMA）功能。

1978~1981年推出第三代16位微处理器，从各项性能指标看，第三代微处理器比第二代微处理器的性能提高了很多，已达到或超过原来的中低档小型机的水平。

1985年，Intel公司推出了32位微处理芯片80386。80386有两种结构：80386 SX和80386 DX。

1990年，Intel公司在80386的基础上研制出新一代32位微处理芯片80486，其性能比80386大大提高，并采用了RISC（Reduced Instruction Set Computing，精简指令集计算）设计思想。

1993年3月，Intel公司推出了第五代微处理芯片Pentium（简称P5），它的外部数据总线为64位，内部总线为32位，工作频率为66MHz。次年，Intel公司又推出了第2代Pentium，在体系结构上采用了RISC技术，可以说它是CISC和RISC技术相结合的产物。

1995年2月，Intel公司发布代号为P6（Pentium Pro）的新一代微处理器产品。

1998年到1999年，Intel公司又推出了Pentium的改进型，即Pentium II和Pentium III。

到了2001年底Intel公司又推出了代号为Northwood的Pentium 4，很快成为主流高端32位CPU市场的佼佼者，特别是在多媒体应用领域中，更具有突出的表现。同时，与Intel公司进行竞争的AMD公司，也不断推出新型微处理器。

在不断完善32位CPU系列的同时，Intel公司和AMD公司在开发第7代CPU即64位CPU方面展开了激烈的竞争，并采取了不同的策略。Intel公司在开发64位的Itanium（安腾）时，放弃一直沿用的x86架构，而在IA-64架构的体系中采用了所谓的EPIC（Explicitly Parallel Instruction Computing，显式并行指令计算）核心技术，保持了技术上的优势。而AMD公司在其开发64位CPU K8 SledgeHammer（大锤）时，则采取了更为平滑的过渡方式，尽管它在运行64位软件时速度不及Intel公司的Itanium，但由于它注重增强同IA-32指令的兼容性，使其在执行IA-32软件时速度明显高于Itanium。

1.1.2 常用术语

（1）位（bit）：是计算机所能表示的最小的数据单位。每一位只能有两种状态：“0”或“1”。bit是Binary Digit的缩写。

（2）字节（Byte）：一个8位二进制数称为一个字节，在计算机中的基本存储单元内容用字节表示。

（3）字（Word）：是微处理器内部数据传输、处理的基本单位。目前PC机常将2个字节定义为一个字，即一个字为16位二进制数。

（4）指令：用二进制代码组成，规定微处理器进行某种操作的命令。即由“0”和“1”编码组成的代码，微处理器每一个操作均对应一组惟一的编码。

（5）指令系统：是指一台计算机所能识别的全部指令。不同类型的微处理器，其指令系统不同，一般不能兼容。如：IBM机、苹果机、各种单片机的指令系统均不相同。

（6）程序：指令的有序集合。即：为使计算机完成某种工作而编制的一系列指令。计算机将按照这一系列指令一步一步地工作。而这一系列指令就组成程序。

（7）存储器：用于存储数据和程序。而数据和程序均用二进制编码表示。我们把存储器看成很多很多的单元，每个基本存储单元存储8位二进制编码（称为一个字节），并且均有自己的编号，这个编号称为该存储单元的地址。

（8）存储容量单位：

1B=8bit (称为一个字节)

1KB=1024B=1024×8bit

1MB=1024KB=1024×1024B

1GB=1024MB=1024×1024×1024B

1TB=1024GB=1024×1024×1024×1024B

(9) 地址: 指存储单元或 I/O 接口的编号。每个存储单元或 I/O 接口均有自己的惟一的物理编号, 即称为物理地址。

1.1.3 正文中使用的符号

R	寄存器	M	存储器
R8	8 位寄存器 (字节寄存器)	M8	8 位存储器 (字节单元)
R16	16 位通用寄存器 (字寄存器)	M16	16 位存储器 (字单元)
R32	32 位通用寄存器 (双字寄存器)	M32	32 位存储器 (双字单元)
src	源操作数	mem16	16 位存储器单元的逻辑地址
dest	目标操作数	mem32	32 位存储器单元的逻辑地址
n_8	8 位立即数	EA	有效地址
n_{16}	16 位立即数	disp	位移量
n_{32}	32 位立即数	disp8	位移量为 8 位二进制数补码
n_{64}	64 位立即数	disp16	位移量为 16 位二进制数补码

1.2 计算机中数与编码的表示方法

计算机的工作就是处理数据或编码, 那么, 计算机是如何描述数和编码的? 根据冯·诺伊曼计算机的概念, 计算机中采用二进制数表达。因此, 计算机中无论是数还是编码均用二进制数描述。但是为了描述方便, 在编程中常采用不同的进制来描述数据。如: 为了书写方便且不易出错, 常用十六进制数表示, 为顾及日常生活的习惯常使用十进制数 (BCD 码), 在大型机、巨型机中还使用八进制数等。本书中常用的数制有: 二进制数、十进制数、十六进制数。了解二、十、十六进制数的表达方式及相互转换的方法是非常重要的。除此之外, 还应了解计算机中如何描述有符号数和编码。

1.2.1 进制表示

数制即为表示一个数的格式, 目前均以进位制来描述, 即每位的数值为该位的数字乘以该位的权值, 每位之间均为一个进位, 即:

$$N_r = d_{n-1} \times r^{n-1} + d_{n-2} \times r^{n-2} + \dots + d_1 \times r^1 + d_0 \times r^0 + d_{-1} \times r^{-1} + \dots + d_{-m} \times r^{-m}$$

其中, r 为进位值; n 为整数位数; m 为小数位数; d_i 为第 i 位的数字 (r 个数字中之一)。

(1) 十进制数, 即 $r=10$, 为逢十进一, 借一当十, 需要十个数字符号 (0~9), 第 i 位以 10^i 为权, 如: $N_{10} = (683.74)_{10} = 683.74D = 6 \times 10^2 + 8 \times 10^1 + 3 \times 10^0 + 7 \times 10^{-1} + 4 \times 10^{-2} = 683.74$ 。

(2) 二进制数, 即 $r=2$, 为逢二进一, 借一当二, 需要两个数字符号 0 和 1, 第 i 位以 2^i 为权, 如: $N_2 = (10001110.11)_2 = 10001110.11B = 1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 142.5$ 。

$$2^1+0 \times 2^0+1 \times 2^{-1}+1 \times 2^{-2}=(571.75)_{10}$$

(3) 十六进制数，即 $r=16$ ，为逢十六进一，借一当十六，需要十六个数字符号 0~9 及 A~F，第 i 位以 16^i 为权，如： $N_{16}=(5A2.F1)_{16}=5A2.F1H = 5 \times 16^2+A \times 16^1+2 \times 16^0+F \times 16^{-1}+1 \times 16^{-2}=(1442.941)_{10}$ 。

(4) 各进制之间相互转换规则：一个数可以用不同的进制表示，各进制间转换过程中，整数部分和小数部分的处理方法不同，其转换规则为：

{	十进制数 → 二进制:	{ 整数部分为将数据N的整数部分连续除2取余，直至商为0 { 小数部分为将数据N的小数部分连续乘2取整，直至满足精度要求
	十进制数 → 十六进制:	{ 整数部分为将数据N的整数部分连续除16取余，直至商为0 { 小数部分为将数据N的小数部分连续乘16取整，直至满足精度要求
{	二进制数 → 十进制 十六进制数 → 十进制	按权展开，即： $N = \sum_{i=n-1}^{-m} d_i \cdot r^i$ 。

其中， n 为整数位数； m 为小数位数； r 为进制值（如：十六进制数，则 $r=16$ ）； d_i 为第 i 位上的数字（如：十六进制数 → 十进制时， d_i 为 0 到 15 中的某个数字）。

【例 1-1】 将十进制数 87.65 用 8 位二进制整数和 4 位二进制小数表示。

解： 整数部分：

2	87	余数
2	43	1
2	21	1
2	10	1
2	5	0
2	2	1
2	1	0
	0	1

↑ 低位

↑ 高位

小数部分：

	0.	65	整数
x		2	
	1	.30	1
x		2	
	0	.60	0
x		2	
	1	.20	1
x		2	
	0	.40	0

↑ 高位

↓ 低位

所以 $87.65D = 01010111.1010B$

【例 1-2】 将十进制数 3587.658 转换成 4 位十六进制整数和 2 位十六进制小数。

解： 整数部分：

16	3587	余数
16	224	3
16	14	0
	0	E

↑ 低位

↑ 高位

	0.658	
x)	16	
	3948	整数
+)	658	
	10	.528
x)	16	A
	3168	
+)	528	
	8	.448

↑ 高位

↓ 低位

即 3587.658D = 0E03.A8H

【例 1-3】 将十六进制数 57.AH 转换成十进制数。

解: $57.AH = 5 \times 16^1 + 7 \times 16^0 + A \times 16^{-1} = 80 + 7 + 0.625 = 87.625$

注意: 在汇编语言程序中十进制数的标记 D 可以省略不写, 但十六进制数的标记 H 和二进制数的标记 B 不能省略。

1.2.2 定点和浮点表示

在实际应用中, 要处理的数据多数是带有小数点的, 而整数是小数的特殊情况。然而计算机只能识别 0、1 二进制数, 不会识别小数点。对具有小数部分的数, 在计算机中有两种表示方法: 定点数和浮点数。

(1) 定点数: 即小数点固定的数。或定义为纯整数, 或定义为纯小数, 如图 1-1 所示。

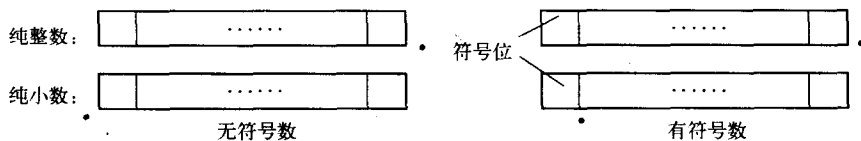


图 1-1 定点数格式示意

(2) 浮点数: 又称为实数, 即小数点浮动的数, 此类数实际上是用指数形式表示, 如: $\pm L \cdot 2^{\pm C}$, 其中 C 称为阶数 (即指数), 其位数多少决定数据表示的范围大小。L 称为尾数, 其位数多少决定数据的精度。用定点数表示浮点数的格式如图 1-2 所示。阶符和尾符分别为阶数和尾数的符号, 0 表示 '+' 号, 1 表示 '-' 号。

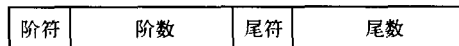


图 1-2 用定点数表示浮点数

【例 1-4】 用国际标准 (IEEE-754) 的 32 位浮点数 (单精度浮点数) 表示 876.5 和 -0.125。

分析: 目前, 大多数编译器都支持国际标准 (IEEE-754) 的格式。其 32 位浮点数的格式要求为, 尾数占 24 位, 其中 23 位 ($D_{22} \sim D_0$) 用绝对值表示有效数据, 1 位 (D_{31}) 为符号位, 阶码占 8 位 ($D_{30} \sim D_{23}$), 如图 1-3 所示。

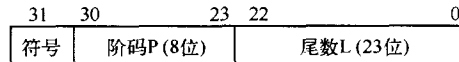


图 1-3 国际标准 (IEEE-754) 的 32 位浮点数格式

如果按照尾数归一化格式要求, 其中二进制数尾数中整数部分一定为 1, 故尾数只写小数部分, 则有 $0.5 \leq |L| < 1$, $P = 7FH \pm$ 阶数 (阶符为 "+" 则加, 阶符为 "-" 则减)。所以有:

$$876.5 = 1101101100.100B = 1.10110110010000000000000B * 2^{1001B}$$

$$\text{尾数 } L = 10110110010000000000000B,$$

$$\text{阶码 } P = 0111\ 1111B + 1001B = 88H$$

876.5 的浮点数表示为 = 0 10001000 101101100100000000000000B = 445B2000H

-0.125 = -0.0010000B = 1.000000000000000000000000B * 2^{-11B}

尾数 L = 000000000000000000000000B,

阶码 P = 0111 1111B-11B = 7CH

-0.0125 的浮点数表示为 = 1 01111100 000000000000000000000000B = BE000000H

从例 1-4 可以看出, 用浮点表示数据比较复杂, 运算也复杂 (需要阶数对齐)。一般由专用的浮点运算部件 (FPU) 实现浮点运算, Pentium 处理器开始将浮点运算部件 (FPU) 集成在微处理器内部, 用专门的浮点运算指令集实现浮点数据处理操作。表 1-1 列出了几种浮点数据的特点。双精度浮点数的阶码 P=3FFH±阶数, 长双精度浮点数的阶码 P=3FFFH±阶数。

表 1-1 几种浮点数据的特点

数据类型	总位数	字节数	符号位	阶码位数	尾数位数	数据范围 (十进制数)	有效数据位数 (十进制数)
单精度浮点数 (float)	32位	4	1位	8位	23位	$1.18 \times 10^{-38} \sim 3.40 \times 10^{38}$	6~7位
双精度浮点数 (double)	64位	8	1位	11位	52位	$2.23 \times 10^{-308} \sim 1.79 \times 10^{308}$	15~16位
长双精度浮点数 (long double)	80位	10	1位	15位	64位	$3.37 \times 10^{-4932} \sim 1.18 \times 10^{4932}$	18~19位

定点数与浮点数各有优缺点, 应用在不同需求的场合:

(1) 定点数表述简单、运算简单、速度快。浮点运算复杂、速度慢。

(2) 在相同字长的计算机中定点数描述数据范围小、精度低, 浮点数描述数据范围大、精度高。

1.2.3 有符号数的表示

计算机也可以处理有符号数, 那么计算机是如何描述、处理有符号数的? 计算机表示有符号数是将符号数值化, 即: 将二进制数的最高位定义为符号位, 用“0”表示“+”号, 用“1”表示“-”号; 无符号数是有符号数的特例, 可以看成正数。

为了计算方便, 引入了原码、反码、补码的概念。同一数的原码、反码和补码, 是从不同的角度表示这个数。

1. 原码

原码即为最高位为符号位 (0 表示正数, 1 表示负数), 其余位均为数值位的二进制数, 所表示的数据范围为: $1-2^{n-1} \sim 2^{n-1}-1$ 。如 8 位二进制数 (n=8), 所表示的数据范围为: $1-2^{8-1} \sim 2^{8-1}-1$, 即-127~+127。

例如: $[+56]_{原} = 00111000B = 38H$

$[-56]_{原} = 10111000B = B8H$

0 的原码有两种表示:

$[+0]_{原} = 00000000B = 00H$

$[-0]_{原} = 10000000B = 80H$

2. 反码

正数的反码与其原码相同。负数的反码等于原码中符号位不变其他各位取反。反码表示