



普通高等教育“十一五”国家级规划教材
大学计算机规划教材



微机原理与 接口技术 (第4版)

◆ 彭 虎 周佩玲 傅忠谦 编著

Microcomputer Principle and
Interface Technology, Fourth Edition

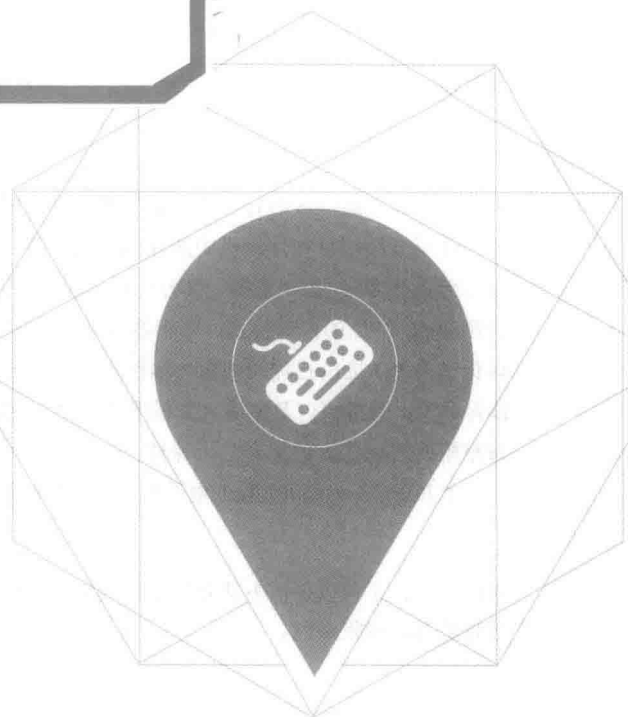
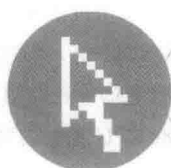


中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

普通高等教育“十一五”
大学计算机规划教材



微机原理与 接口技术（第4版）

◆ 彭 虎 周佩玲 傅忠谦 编著

Microcomputer Principle and
Interface Technology Fourth Edition

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书为普通高等教育“十一五”国家级规划教材。

本书介绍信息在计算机中的存储形式、数制及相互转换、二进制数的算术和逻辑运算等基础知识；软件部分讲述 8086 指令系统、部分伪指令和 DOS 功能调用及汇编语言程序设计和调试的全过程；硬件部分介绍 8086 CPU 的内部特点、寄存器及相关概念、存储器的分类及层次结构、物理地址形成、译码电路等；讨论诸多 I/O 接口芯片的结构、编程及应用，在串行通信中还介绍了 USB 总线；讨论并举例说明了 A/D、D/A 芯片、微机接口及应用，本书还对 80286、80386 CPU 主要内容及其体系做了简要介绍。全书共 12 章，每章附有习题，提供配套电子课件。

本书适合作为高等院校非电类理工科学生相关课程的教材，也可以作为相关技术人员或爱好者的参考书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目 (CIP) 数据

微机原理与接口技术 / 彭虎, 周佩玲, 傅忠谦编著. —4 版. —北京: 电子工业出版社, 2016.6
ISBN 978-7-121-28983-5

I. ① 微… II. ① 彭… ② 周… ③ 傅… III. ① 微型计算机—理论—高等学校—教材 ② 微型计算机—接口技术—高等学校—教材 IV. ① TP36

中国版本图书馆 CIP 数据核字 (2016) 第 125942 号

策划编辑: 章海涛

责任编辑: 章海涛

印 刷: 涿州市京南印刷厂

装 订: 涿州市京南印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 20.25 字数: 554 千字

版 次: 2007 年 12 月第 1 版

2016 年 6 月第 4 版

印 次: 2016 年 6 月第 1 次印刷

定 价: 42.00 元



凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888，88258888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：192910558 (QQ 群)。

第 4 版前言

2010 年,教育部启动“卓越工程师教育培养计划”,旨在培养造就一大批创新能力强、适应经济社会发展需要的高质量各类型工程技术人才。“卓越计划”具有三个特点:一是行业企业深度参与培养过程;二是学校按通用标准和行业标准培养工程人才;三是强化培养学生的工程能力和创新能力。显然,强化培养学生的工程能力和创新能力的一个途径就是改革教材,把书本上的知识,特别是工程技术方面的教材内容与实际结合起来,把提出问题、分析问题、解决问题的方法带入到书本的讲解中,使学生在 学习中领会其工程思想,把握其工程特点,自觉地把所学知识高水平地应用到工作中。

我们在总结 20 多年的微机教学经验基础上,于 2006 年出版了相关教材,对计算机的软硬件的充满工程色彩的描述和讲解是本书的主要特色。本书主要在中国科学技术大学非电子系及专业使用,原书写于 2006 年,第 3 版于 2011 年出版,累计印刷了 10 次,超过 4 万册,显示了读者和有关教学部门对此书的重视和厚爱。

根据广大读者和从事微机教学同人的意见,并参考笔者历年来的讲稿,本书进行了修订,修改内容包括:① 删去了一些不必要的内容;② 更正了原书中的一些错误;③ 对书中一些概念作了进一步说明;④ 增加了一些实例。具体来讲,在软件部分对一些重要指令在概念上、在编程使用上进一步进行了描述。为了提高读者的编程能力,本书增加了一些新的编程例子,并对程序尽量详细地给予了说明。

针对本书的讲课安排,笔者建议总学时在 60~80 学时,主要授课范围是第 1~10 章,各单位可以根据自己的教学安排对其内容进行有重点的讲授。第 11 章和 12 章是微机原理知识的扩展,对非电类的学生不一定讲授。

笔者从事微机教学已近 30 年,虽积累了许多教学经验,但不敢说本教材能尽如人意,微机原理与接口内容很庞杂,书中的不妥之处在所难免,今后我们会继续努力,完善这部教材。

本书由彭虎、周佩玲、傅忠谦进行修订编写,彭虎负责全书的统稿。

由于作者学识浅显,经验有限,书中难免会出现许多不足和遗漏,希望各位同行批评指正。

本书为教学老师提供相关教学课件,可从网站 <http://www.hxedu.com.cn> 下载。

读者反馈:192910558 (QQ 群)。

作 者

中国科学技术大学信息技术学院
合肥工业大学医学工程学院

目 录

第 1 章 计算机基本知识	1
1.1 微型计算机组成	1
1.2 微型计算机中信息的表示及运算基础	2
1.2.1 二进制数的表示和运算	3
1.2.2 二-十进制 (BCD) 数的表示和运算	4
1.2.3 十六进制数的表示和运算	5
1.2.4 带符号二进制数的表示和运算	6
1.2.5 字符的编码表示	9
1.3 几种进制之间的相互转换	10
1.3.1 十进制整数到任意进制整数的转换	10
1.3.2 任意进制整数到十进制整数的转换	11
1.3.3 二进制数与十六进制数的转换	12
1.3.4 带符号二进制整数与十进制整数的转换	12
习题 1	13
第 2 章 8086 系统结构	14
2.1 8086 CPU 结构	14
2.1.1 8086 CPU 的内部结构	14
2.1.2 8086 CPU 的寄存器结构	16
2.1.3 8086 CPU 的引脚及功能	20
2.2 8086 系统的结构和配置	22
2.2.1 8086 存储器结构	23
2.2.2 8086 的输入/输出结构	26
2.2.3 8086 的最小模式和最大模式系统	27
2.3 8086 CPU 内部时序	30
习题 2	34
第 3 章 8086 指令系统	35
3.1 8086 指令的特点	35
3.2 8086 的寻址方式	36
3.2.1 8086 寻址方式的说明	36
3.2.2 寻址方式介绍	37
3.3 8086 的指令格式及数据类型	40
3.4 8086 的指令集	42
3.4.1 数据传输指令	42
3.4.2 算术运算指令	46

3.4.3 位操作指令	52
3.4.4 串处理指令	57
3.4.5 程序控制转移指令	59
3.4.6 处理器控制指令	65
习题 3	66
第 4 章 8086 汇编语言程序设计	70
4.1 8086 汇编语言的语句	70
4.2 8086 汇编语言中的伪指令	71
4.2.1 符号定义语句	72
4.2.2 变量定义语句	72
4.2.3 段定义语句	74
4.2.4 过程定义语句	76
4.2.5 结束语句	77
4.3 8086 汇编语言中的运算符	77
4.3.1 常用运算符和操作符	77
4.3.2 运算符的优先级别	79
4.4 汇编语言程序设计	80
4.4.1 汇编语言程序设计基本步骤	80
4.4.2 汇编语言程序的基本结构	80
4.5 宏定义与宏调用	86
4.6 汇编语言程序设计与上机调试	88
4.6.1 汇编语言程序设计实例	88
4.6.2 DOS 功能调用与子程序设计	97
4.6.3 汇编语言程序上机调试	101
习题 4	102
第 5 章 存储器原理与接口	103
5.1 存储器分类	103
5.2 多层存储结构	105
5.3 主存储器及存储控制	107
5.3.1 主存储器	107
5.3.2 主存储器的基本组成	108
5.4 8086 系统的存储器组织	111
5.4.1 8086 CPU 的存储器接口	111
5.4.2 存储器接口举例	113
5.5 现代内存芯片技术	118
习题 5	118
第 6 章 微型计算机的输入/输出	119
6.1 CPU 与外设通信的特点	119
6.1.1 I/O 端口的寻址方式	119
6.1.2 I/O 端口地址的形成	120

6.2 输入/输出方式	121
6.3 CPU 与外设通信的接口	122
6.3.1 同步传输方式与接口	122
6.3.2 异步查询方式与接口	123
6.4 8086 CPU 的输入/输出	126
习题 6	127
第 7 章 可编程接口芯片	128
7.1 可编程并行接口芯片 8255A	129
7.1.1 8255A 的内部结构	129
7.1.2 8255A 的引脚	130
7.1.3 8255A 的工作方式及编程	131
7.1.4 8255A 工作方式的功能	133
7.1.5 8255A 应用举例	140
7.2 可编程定时/计数器接口芯片 8253	146
7.2.1 8253 的内部结构	147
7.2.2 8253 的引脚分配	149
7.2.3 8253 的编程	149
7.2.4 8253 的工作方式	152
7.2.5 8253 的工作方式小结	158
7.2.6 8253 应用举例	159
习题 7	162
第 8 章 串行输入/输出接口	164
8.1 串行通信接口	164
8.1.1 串行通信的实现	165
8.1.2 串行通信的基本概念	170
8.1.3 可编程串行通信接口芯片 8251A 简介	173
8.1.4 串行通信接口 RS-232C	181
8.2 USB 总线简介	187
8.2.1 USB 概述	187
8.2.2 USB 工作原理	190
8.2.3 USB 传输方式	192
8.2.4 USB 设备列举	193
8.3 USB 总线转接芯片——CH341 简介	194
习题 8	196
第 9 章 中断和中断管理	197
9.1 中断原理	197
9.1.1 从无条件传输、条件传输到中断传输	197
9.1.2 中断概念	198
9.1.3 中断应用	199
9.2 中断系统组成及其功能	200

9.2.1	与中断有关的触发器	200
9.2.2	中断条件	201
9.2.3	中断响应过程	202
9.3	中断源识别及中断优先权	203
9.3.1	中断源识别	203
9.3.2	中断优先权	206
9.4	8086 中断系统	207
9.4.1	不可屏蔽中断	208
9.4.2	可屏蔽中断	208
9.4.3	软件中断	208
9.4.4	中断概念的再讨论	211
9.5	8086 CPU 的中断管理	211
9.5.1	8086 CPU 的中断处理顺序	211
9.5.2	8086 CPU 的中断服务入口地址表	211
9.5.3	中断入口地址设置	212
9.6	可编程中断控制器 8259A 简介	215
9.6.1	8259A 的内部结构及引脚分配	215
9.6.2	8259A 的中断管理方式	217
9.6.3	8259A 的编程与应用	219
9.7	IBM PC 硬件中断	223
9.7.1	中断设置	224
9.7.2	计算机中断资源的使用	224
9.7.3	中断举例	225
	习题 9	227
第 10 章	D/A 转换和 A/D 转换接口及其应用	228
10.1	从物理信号到电信号的转换	229
10.2	D/A 转换器及其接口技术	231
10.2.1	并行 8 位 D/A 转换芯片 AD558 及其接口	232
10.2.2	串行 8 位 D/A 转换器 TLC5620	233
10.2.3	12 位 D/A 转换器及接口	237
10.3	A/D 转换器及其接口技术	237
10.3.1	A/D 转换原理	237
10.3.2	A/D 转换与微机接口技术的一般原理	238
10.3.3	A/D 转换与微机接口电路	239
10.3.4	A/D 转换芯片 ADC0809	241
10.3.5	串行 8 位 A/D 转换器 TLC0831	245
10.4	微机应用实例	247
	习题 10	249
第 11 章	80286 微处理器	250
11.1	80286 微处理器基本原理概述	250

11.1.1	80286 内部结构简介	250
11.1.2	80286 芯片引脚功能	254
11.1.3	80286 支持的数据类型和指令系统	255
11.1.4	80286 的存储器管理	256
11.1.6	保护虚地址方式下存储器管理	257
11.2	80286 的系统配置	262
习题 11		264
第 12 章	80386 微处理器	265
12.1	80386 系统结构	265
12.1.1	80386 微处理器的基本结构	265
12.1.2	80386 的寄存器组成	268
12.1.3	80386 的存储器管理	270
12.1.4	80386 的保护机制	275
12.1.5	80386 系统组成	275
12.2	80386 的指令系统	277
12.2.1	80386 的寻址方式	277
12.2.2	80386 的指令系统	277
12.3	80386 的存储器扩展	280
12.3.1	基本的存储器接口控制线	281
12.3.2	基本的存储器接口	281
12.3.3	高速缓冲存储器	284
12.3.4	Intel 82385 高速缓存控制器	288
12.4	80386 的输入/输出接口	289
12.4.1	80386 的 I/O 系统	289
12.4.2	多功能接口芯片 82380	291
12.5	80386 的异常和中断及其处理	294
12.5.1	实模式下异常和中断的处理过程	295
12.5.2	保护虚地址方式下异常和中断的处理过程	295
12.6	精简指令集计算机 (RISC) 简介	298
12.6.1	RISC 的基本原理	298
12.6.2	RISC 的特色及其难点所在	300
12.6.3	RISC 的关键技术	301
12.7	80x86 典型微处理机介绍	303
12.7.1	80486 CPU	303
12.7.2	Pentium 系列微处理机	308
习题 12		310
参考文献		311

第 1 章 计算机基本知识

本章导读

- ☆ 微型计算机组成
- ☆ 微型计算机中信息的表示
- ☆ 微型计算机中信息的运算基础
- ☆ 几种进制之间的相互转换

电子计算机是 20 世纪科学技术最卓越的成就之一，它的飞速发展在人类科技史上还没有与之相提并论的。计算机技术的发展所带来的信息技术的飞速发展，给人类社会带来了进步，使人们的生产、生活发生了深刻的变化。

计算机在现代科学技术的发展中起着越来越重要的作用。多媒体技术、计算机网络技术、智能信息处理技术、自适应控制技术、数据挖掘与处理技术、机械设计 CAD、金融电子等都离不开计算机。

1.1 微型计算机组成

1946 年，世界上第一台电子计算机由美国宾夕法尼亚大学研制成功。尽管它重达 30 吨，占地 170 m²，耗电 140 kW，用了 18 800 多个电子管，每秒钟仅能做 5000 次加法，但美国陆军用它计算弹道比人工计算效率提高 8400 倍。当时是用改变线路连接的方法来编排程序的，因此每解一道题都要依靠人工改接线路，准备时间大大超过实际计算时间，所以还称不上是自动计算机。

与此同时，第一台电子计算机研制时的顾问冯·诺依曼（Von Neumann）教授和他的同事们提出了以二进制和程序存储控制为核心的通用电子数字计算机体系结构原理，确立了计算机的五个基本部件：输入设备、输出设备、运算器、存储器和控制器，从而奠定了当代电子数字计算机体系结构的基础。现在的微型计算机就是采用这种结构，如图 1-1 所示。

CPU（Central Processing Unit，中央处理单元或微处理器）：包括运算器和控制器，主要功能是让存储器中的程序被逐条地执行所指定的操作。CPU 是微型计算机的核心部件。从 20 世纪 70 年代初诞生第一片 CPU 以来，基本上每两三年就有更新产品出现。

存储器：其主要功能是存放程序和数据。CPU 在工作时，先从存储器取程序，再执行程序，从而完成对数据的处理。

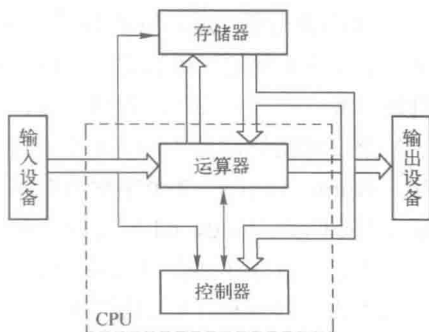


图 1-1 微型计算机的组成

外部设备：用户与机器之间的桥梁，包括输入设备和输出设备。输入设备的功能是，把用户要求计算机处理的数据、文字、图形和程序等各种形式的信息转换为计算机能接受的编码形式，存入到计算机中。输出设备的功能是，把计算机的处理结果以用户需要的形式（如屏幕显示、文字打印、图形图表、语言音响等）输出出来。

如果要求图 1-1 所示系统自动完成解题任务，必须事先将问题分解成计算机能够处理的各步骤，用某种语言将这些步骤描述出来，然后让计算机按规定的步骤控制计算机工作。这就是计算机设计语言，分为低级语言和高级语言。

低级语言有机器语言和汇编语言，前者就是 0、1 码语言，是计算机唯一能够理解且直接执行的语言。用户编写程序时，命令（指令）、数据和其他信息均以二进制编码书写，难读、难懂、难记、难查错、无法交流，给程序设计和计算机的推广、应用、开发等带来许多困难。对机器语言进行改进的第一步是用一些助记符号代替用 0 和 1 描述的某种机器的指令系统，如八进制数、十六进制数或英语单词的缩写等，称为机器语言的助记符号形式（或符号语言）。汇编语言就是在此基础上完善起来的，改善了机器语言的可读性、可记性，汇编语言指令与机器语言指令一一对应。汇编语言是能够利用计算机所有硬件特征且能直接控制硬件的一种程序设计语言，是计算机能够提供给用户的最快且最有效的编程语言，要求程序设计者必须掌握计算机的硬件知识，这对那些仅对问题感兴趣的用户无疑是一个极大的障碍。

面向问题的程序设计语言称为高级语言。用户面向的是自己领域内的问题，如数值计算、工业控制、专家系统、数据管理和数据库等。这些语言属于过程化语言，要求程序员为每个应用任务写出完成该任务的一系列明确的过程，如适用于数值计算的 FORTRAN 语言、适用于商用和行政管理的 COBOL 语言、适用于专家系统使用的 PROLOG 语言等，以及 BASIC、Pascal、C 语言等。用高级语言编写的程序称为源程序，必须通过编译或解释、连接等步骤才能被计算机处理。

1.2 微型计算机中信息的表示及运算基础

本节讨论计算机是如何存储信息的，计算机内部数据是怎样表示的。进行汇编语言程序设计，掌握这些基本知识非常必要。目前使用的计算机是一种电设备，只认识电的信号，如电平的高与低、电路的通与断、晶体管的导通与截止、电子开关的开与关等。将这两种状态用 0 和 1 两个符号表示，0 或 1 就是二进制数的一位，称为比特（bit）。因此在计算机中，任何信息都必须用 0 和 1 的数字组合形式来表示。也就是说，计算机存储和处理的仅仅是二进制信息。

1 个二进制位称为 1 bit；8 个二进制位称为 1 Byte，也称为 1 字节（8 位）；2 个字节称为 1 个字（Word，16 位）；2 个字称为双字（Dword 或 Double Word，32 位）；4 个连续的字称为四字（Qword 或 Quad Word，64 位）；连续的 10 字节称为十字节，它是一个 80 位二进制的值。

人类最初使用十进制计数系统是因为人们有 10 个手指头。计算机却只认识 0 和 1 两个符号，这导致了人与计算机之间的通信问题。发明 ASCII（美国标准信息交换码）就是为了代表打字机键盘中的所有符号。几种数值代码的发明也是为了用计算机能够懂得的形式表示数字值。2 的补码这种数字编码使计算机能够表示整个数字的正和负。汉字编码是为解决用西文键盘输入中文的问题而研究的一种编码。对于汇编语言程序设计，最重要的数字编码是十六进制编码，十六进制编码大大简化了二进制编码的表示，缩短了书写的长度。

1.2.1 二进制数的表示和运算

1. 二进制数的表示

二进制数仅有两个计数符号：0，1。一个8位的二进制数由8个0或1组成，如11010010，计数符号在不同位置有不同的位权。例如：

$$11010010 = 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

我们习惯于在二进制数的后面加上字母B（Binary），如11001101B、10011B。

2. 二进制数的运算

(1) 算术运算

加法规则：“逢2进1”。

$$\begin{array}{r} 0 \quad 0 \quad 1 \quad 1 \\ + 0 \quad + 1 \quad + 0 \quad + 1 \\ \hline 0 \quad 1 \quad 1 \quad 10 \end{array}$$

减法规则：“借1当2”。

$$\begin{array}{r} 0 \quad 1 \quad 1 \quad 10 \\ - 0 \quad - 0 \quad - 1 \quad - 1 \\ \hline 0 \quad 1 \quad 0 \quad 1 \end{array}$$

乘法规则：1与1乘为1，其他为0。

$$\begin{array}{r} 0 \quad 0 \quad 1 \quad 1 \\ \times 0 \quad \times 1 \quad \times 0 \quad \times 1 \\ \hline 0 \quad 0 \quad 0 \quad 1 \end{array}$$

(2) 逻辑运算

逻辑非（NOT）运算：

$$0 \rightarrow 1 \quad 1 \rightarrow 0$$

逻辑与（AND）运算：

$$\begin{array}{r} 0 \quad 0 \quad 1 \quad 1 \\ \wedge 0 \quad \wedge 1 \quad \wedge 0 \quad \wedge 1 \\ \hline 0 \quad 0 \quad 0 \quad 1 \end{array}$$

逻辑或（OR）运算：

$$\begin{array}{r} 0 \quad 0 \quad 1 \quad 1 \\ \vee 0 \quad \vee 1 \quad \vee 0 \quad \vee 1 \\ \hline 0 \quad 1 \quad 1 \quad 1 \end{array}$$

逻辑异或（XOR）运算，又称为“模2和”运算：

$$\begin{array}{r} 0 \quad 0 \quad 1 \quad 1 \\ \vee 0 \quad \vee 1 \quad \vee 0 \quad \vee 1 \\ \hline 0 \quad 1 \quad 1 \quad 0 \end{array}$$

【例 1-1】00011010B + 01101101B = 10000111B。

$$\begin{array}{r} 00011010 \\ + 01101101 \\ \hline 10000111 \end{array}$$

【例 1-2】 $10011011\text{B} - 00110101\text{B} = 01100110\text{B}$ 。

$$\begin{array}{r} 10011011 \\ - 00110101 \\ \hline 01100110 \end{array}$$

【例 1-3】 $01110101\text{B} \times 00110110\text{B} = 0001100010101110\text{B}$ 。

$$\begin{array}{r} 01110101 \\ \times 00110110 \\ \hline 01110101 \\ 01110101 \\ 01110101 \\ + 01110101 \\ \hline 0001100010101110 \end{array}$$

【例 1-4】 $10111001\text{B} \div 1011\text{B} = 00010000\text{B}$ ，余 00001001B 。

$$\begin{array}{r} 00010000 \\ 1011 \overline{) 10111001} \\ \underline{-1011} \\ 00001001 \end{array}$$

【例 1-5】 $10011101\text{B} \wedge 01101110\text{B} = 00001100\text{B}$ 。

$$\begin{array}{r} 10011101 \\ \wedge 01101110 \\ \hline 00001100 \end{array}$$

【例 1-6】 $10011101\text{B} \vee 01101110\text{B} = 11111111\text{B}$ 。

$$\begin{array}{r} 10011101 \\ \vee 01101110 \\ \hline 11111111 \end{array}$$

【例 1-7】 $10011101\text{B} \nabla 01101110\text{B} = 11110011\text{B}$ 。

$$\begin{array}{r} 10011101 \\ \nabla 01101110 \\ \hline 11110011 \end{array}$$

1.2.2 二十进制 (BCD) 数的表示和运算

1. 二十进制数的表示

十进制数有 10 个计数符号：0~9，而计算机仅认识 2 个符号：0、1，因此十进制数的 10 个计数符号需要改用 0 和 1 两个符号的编码表示。10 个符号必须用 4 位二进制编码表示：

0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

4 位二进制编码的其他组合不用。这种用二进制编码表示的十进制数称为 BCD (Binary Coded

Decimal, 二-十进制) 数。

2. 二十进制数的加、减运算

BCD 数的运算规则遵循十进制数的运算规则“逢 10 进 1”。但计算机在进行这种运算时会出
现潜在的错误。

【例 1-8】	BCD 数	十进制数
	1000	8
	+ 0101	+ 5
	1101	13

【例 1-9】	BCD 数	十进制数
	1001	9
	+ 0111	+ 7
	1 0000	16

例 1-8 的两个 BCD 数相加后, 其结果已不是 BCD 数; 而例 1-9 的运算结果不对。究其原因: 在计算机中, 用 BCD 可以表示十进制数, 但其运算规则还是按二进制数进行的。因此, 4 位二进制数相加要 16 才会进位, 而不是逢十进位。

为了解决BCD数的运算问题,采取调整运算结果的措施。调整规则为:当BCD数加法运算结果的4位二进制超过1001(9)或个位向十位有进位时,则加0110(6)进行调整;当十位向百位有进位时,加01100000(60)调整。这是人为地干预进位。

在汇编语言程序设计时，只要用一条指令就可实现，我们称为十进制加法和减法调整指令。其减法调整规则类似。

【例 1-10】 $10001000 \text{ (BCD)} + 01101001 \text{ (BCD)} = 10101011 \text{ (BCD)}$ 。

$$\begin{array}{r} 10001000 \\ + 01101001 \\ \hline 11110001 \\ + 01100110 \quad \leftarrow \text{调整} \\ \hline 1\ 01010111 \end{array}$$

└───────────┘ 进位

【例 1-11】 $10001000 \text{ (BCD)} - 01101001 \text{ (BCD)} = 00011001 \text{ (BCD)}$ 。

$$\begin{array}{r} 10001000 \\ - 01101001 \\ \hline 00011111 \\ - \quad 0110 \quad \leftarrow \text{调整} \\ \hline 00011001 \end{array}$$

1.2.3 十六进制数的表示和运算

1. 十六进制数的表示

十六进制数有 16 个计数符号：0~9 和 A~F。4 个二进制位共 16 种组合状态，这样每个十六进制数的计数符号可对应 4 位二进制数的一种组合状态；反之，一个十六进制符号可以替代一

种 4 位二进制数的组合状态。在阅读和编写汇编语言程序时，经常用十六进制数表示数据、存储单元地址或代码等。表 1-1 列出了十进制数、二进制数、二-十进制数、十六进制数之间的关系。

表 1-1 十进制数、二进制数、二-十进制数、十六进制数之间的关系

十进制数 (D)	二进制数 (B)	二-十进制数 (BCD)	十六进制数 (H)
0	0000	0000	0
1	0001	0001	1
2	0010	0010	2
3	0011	0011	3
4	0100	0100	4
5	0101	0101	5
6	0110	0110	6
7	0111	0111	7
8	1000	1000	8
9	1001	1001	9
10	1010	×	A
11	1011	×	B
12	1100	×	C
13	1101	×	D
14	1110	×	E
15	1111	×	F

在书写数据时，为了区分不同进制的数据，在十进制数后加字母 D 或省略；在二进制数后加字母 B；在十六进制数后加字母 H，对于字母开头的十六进制数，还须在数据前加个 0，以表明它是十六进制数而不是其他，如 25H，43，57D，0A5H，0CD3BH。

说明：采用十六进制主要是缩短二进制数的表示长度，方便程序书写和阅读，在计算机内的操作仍然是二进制数的形式。

2. 十六进制数的加、减运算

加法运算：“逢 16 进 1”。例如：

1

+

9

A

2

+

A

C

3

+

8

B

A

+

5

F

E

+

C

1A

7

+

9

10

9

+

9

12

F

+

1

10

FF

+

1

100

减法运算：“借 1 当 16”。例如：

12

-

5

7

2B

-

12

19

D4

-

6B

69

58

-

E

4A

AB

-

37

74

17

-

9

8

10

-

2

8

1.2.4 带符号二进制数的表示和运算

“+”和“-”在计算机中只能用 0 和 1 表示，我们用 0 表示“+”，用 1 表示“-”，并且符号放在最高有效位。一个 8 位的二进制数表示一个带符号数，最高有效位 D₇ 为符号位。如 +1 表示为 00000001B，+127 表示为 01111111B，-1 表示为 10000001B，-127 表示为 11111111B。

用此方法表示的带符号数，在进行运算时遇到了问题：+1 加 -1 的运算是最明显的例子。

$$\begin{array}{r}
 (+1) \quad 00000001 \\
 + (-1) \quad + 10000001 \\
 \hline
 0 \quad 10000010 \quad (-2)
 \end{array}$$

而+127加-127会得出下面的错误结果：

$$\begin{array}{r}
 (+127) \quad 01111111 \\
 + (-127) \quad +11111111 \\
 \hline
 0 \quad 1 \quad 01111110
 \end{array}$$

符号参与运算

计算机的数据位长度是固定的，如4位、8位、16位、32位、64位等，当最高有效位作为符号后，其数值的大小就减小一半。而且计算机在进行运算操作时，符号也作为数据参与运算。

1. 带符号数的2的补码表示法

带符号数采用上面的编码方法不可取。考虑到计算机的数据位长度一定，因而可用补数的编码形式表示带符号数。在进行运算时，丢弃进位，就可得到正确结果。

对一个正的二进制数的每位求反再加1，即可得在机器中表示的该数的负数，即2的补码表示法。在这种编码方式中，正数的补码就是该正数。下面以8位二进制数为例，求负数的补码。

【例 1-12】

$$\begin{array}{r}
 (+1) \quad 00000001 \\
 \text{逐位求反} \quad 11111110 \\
 \text{加 1} \quad + \quad 1 \\
 \hline
 -1 \quad 11111111
 \end{array}$$

【例 1-13】

$$\begin{array}{r}
 (+45H) \quad 01000101 \\
 \text{逐位求反} \quad 10111010 \\
 \text{加 1} \quad + \quad 1 \\
 \hline
 (-45H) \quad 10111011
 \end{array}$$

用2的补码对带符号数进行编码，是目前计算机中常用的方法。表1-2中列出了部分带符号数2的补码值。

表 1-2 带符号数2的补码值表（8位）

十进制数	十六进制数 (H)	2 的补码值 (B)	十进制数	十六进制数 (H)	2 的补码值 (B)
+127	7F	01111111	0	0	00000000
+100	64	01100100	-1	FF	11111111
+3	3	00000011	-2	FE	11111110
+2	2	00000010	-100	9C	10011100
+1	1	00000001	-128	80	10000000

2. 带符号数的运算

带符号数在计算机中用其补码表示后，其运算方法与二进制数的运算相同，也会出现其特有的问题。下面通过例题来分析问题并且讨论如何解决这些问题。

【例 1-14】 用上面的编码方法完成(+1)+(-1)的运算。

$$\begin{array}{r}
 (+1) \quad 00000001 \\
 + (-1) \quad + 11111111 \\
 \hline
 0 \quad 10000000
 \end{array}$$

进位丢弃

【例 1-15】完成(+65H)+(-23H)的运算。

$$\begin{array}{r}
 (+65H) \quad 01100101 \\
 + (-23H) \quad + 11011101 \\
 \hline
 +42H \quad 101000010
 \end{array}$$

进位丢弃

从以上例题可知，减法可用加上该数的补码，丢弃进位运算完成。

【例 1-16】完成(+5EH)+(+37H)运算。

$$\begin{array}{r}
 (+5EH) \quad 01011110 \\
 + (+37H) \quad +00110111 \\
 \hline
 +95H \quad 10010101
 \end{array}$$

符号位变为“1”

【例 1-17】完成(-62H)+(-3BH)运算。

$$\begin{array}{r}
 (-62H) \quad 10011110 \\
 + (-3BH) \quad + 11000101 \\
 \hline
 -9DH \quad 101100011
 \end{array}$$

符号位变为“0”
进位丢弃

在例 1-16 和例 1-17 中，两正数相加后，其结果的符号变为“1”，即两正数相加得出结果为负数；两负数相加后，其结果的符号变为“0”。这显然是错误的。因为带符号数的符号占用了 1 个二进制数的最高有效位，当运算结果数值超出二进制数所能表达的范围时，其值就进入符号位，因而可能改变其运算结果的符号，得到错误结果。以上两例是 8 位二进制数运算，从表 1-2 可知：用 8 位二进制数表示带符号数，数值范围在+127（补码为 7FH）到-128（补码为 80H）之间。

例 1-16 的运算结果为 95H，运算结果进入了符号位；例 1-17 中运算结果为-9DH，运算结果也进入了符号位。因此这两例运算均得不到正确结果。在这种情况下，运算后的值不可取。因为运算结果超出了数值范围，故称为“溢出”。

【例 1-18】完成(-1FH)+(-4AH)运算。

$$\begin{array}{r}
 (-1FH) \quad 11100001 \\
 + (-4AH) \quad + 10110110 \\
 \hline
 -69H \quad 110010111
 \end{array}$$

符号位不变
进位丢弃

【例 1-19】完成(+6EH)+(-7CH)运算。

$$\begin{array}{r}
 (+6EH) \quad 01101110 \\
 + (-7CH) \quad + 10000100 \\
 \hline
 -0EH \quad 11110010
 \end{array}$$

符号位不变

在例 1-18 的二进制数运算中，虽然运算结果数值的最高位（D₆）进入符号位（D₇），但因符