



普通高等教育“十二五”规划教材

汇编语言程序设计

(第二版)

詹仕华 主 编

满正行 张旭玲 副主编



中国电力出版社
CHINA ELECTRIC POWER PRESS



普通高等教育“十二五”规划教材

汇编语言程序设计

(第二版)

主 编 詹仕华

副主编 满正行 张旭玲

编 写 余国伟 刘志都



中国电力出版社

内 容 提 要

汇编语言是各种 CPU 所提供的机器指令的助记符的集合。编程人员可以直接通过汇编语言程序控制硬件系统工作。本书以 8086 指令系统为主,介绍汇编语言程序设计的基本理论和基本方法。本书为普通高等教育“十二五”规划教材。

全书共分为 8 章,第 1 章介绍微型计算机基础知识,第 2~8 章介绍 80x86 微处理器及系统结构,80x86 指令系统,汇编语言程序格式,基本程序设计,应用程序设计,输入、输出和中断程序设计,高级语言与汇编语言混合编程等。本书逻辑性强,层次分明,特别注重理论和实践相结合,有大量应用程序实例,并有配套的习题解答、课程实验和课程实习的辅助教材。

本书可作为普通高等院校计算机科学与技术及其相关专业的汇编语言程序设计课程的教材,也可作为成人函授教育或高职高专相关专业教材,还可作为自学汇编语言程序设计的读者和相关技术人员的参考用书。

图书在版编目(CIP)数据

汇编语言程序设计 / 詹仕华主编. —2 版. —北京: 中国电力出版社, 2015.1

普通高等教育“十二五”规划教材

ISBN 978-7-5123-6961-0

I. ①汇… II. ①詹… III. ①汇编语言—程序设计—高等学校—教材 IV. ①TP313

中国版本图书馆 CIP 数据核字(2014)第 308624 号

中国电力出版社出版、发行

(北京市东城区北京站西街 19 号 100005 <http://www.cepp.sgcc.com.cn>)

北京市同江印刷厂印刷

各地新华书店经售

*

2008 年 2 月第一版

2015 年 1 月第二版 2015 年 1 月北京第五次印刷

787 毫米×1092 毫米 16 开本 17 印张 411 千字

定价 34.00 元

敬告读者

本书封底贴有防伪标签,刮开涂层可查询真伪
本书如有印装质量问题,我社发行部负责退换

版 权 专 有 翻 印 必 究

前 言

本书是编者在第一版的基础上，结合汇编语言程序设计课程的教学经验和软件开发经验进行修订的。本书修订主要考虑如下。

汇编语言是一种面向机器的语言，它能够利用计算机所有硬件特性并能直接控制硬件，所以汇编语言程序设计是计算机科学与技术及其相关专业重要的一门专业基础课程。作为汇编语言程序设计的教材，在讲明基本内容的基础上，应做到重点突出、难点明确，同时强调具体应用，注重理论教学和实践教学相结合。这是本书第一版时编写的思路，因此，在编写第二版时仍保持了第一版的总体思路，对部分内容进行了删除和补充，同时也调整了内容次序，确保内容层次分明，逻辑性更严密。

随着计算机软件技术和硬件技术的发展，汇编语言和高级语言相结合的开发技术已广泛应用于包含系统软件等的软件开发。因此，第二版增加了高级语言与汇编语言混合编程的内容，以便使读者了解高级语言与汇编语言混合编程的基本方法，提高编程能力和软件设计水平。

在修订本书的同时，也修订了与本书配套的《汇编语言程序设计习题解答及课程实验、设计辅导》一书。

本书由满正行编写第1章、第5章和第8章；詹仕华编写第2章、第7章；张旭玲编写第3章和第4章；余国伟编写第6章；刘志都编写附录部分；全书由詹仕华统稿。限于编者水平，书中疏漏与不足之处在所难免，殷切希望广大读者批评指正。

本书的再版，得到了中国电力出版社的大力支持和协助，谨此表示诚挚的谢意。

编 者

2014年12月

第一版前言

为贯彻落实教育部《关于进一步加强高等学校本科教学工作的若干意见》和《教育部关于以就业为导向深化高等职业教育改革的若干意见》的精神，加强教材建设，确保教材质量，中国电力教育协会组织制订了普通高等教育“十一五”教材规划。该规划强调适应不同层次、不同类型院校，满足学科发展和人才培养的需求，坚持专业基础课教材与教学急需的专业教材并重、新编与修订相结合。本书为新编教材。

汇编语言是一种面向机器的语言，它能够利用计算机所有硬件特性并能直接控制硬件，所以汇编语言是很多相关课程（如微机原理与接口技术、操作系统等）的重要基础。如果想从事计算机科学方面的工作，汇编语言的基础是不可缺少的。因为工作平台、研究对象都是机器。

本书编写过程结合了根据作者自身的教学和实践经验，对教材的知识点、重点和难点有较好的把握，并将实际开发的应用程序编入教材中。注重理论教学和实践教学相结合，通过大量实例分析程序设计的方法，逻辑性强和层次分明，这些将有助于对本教材的理解和程序设计能力的提高。为了便于对教材的理解和加强实践环节，与本书配套编写了《汇编语言程序设计习题解答及课程实验、设计辅导》。

本教材由李应兴编写第1章、第5章和第6章；张旭玲编写第3章和第4章；刘志都编写附录部分；詹仕华编写第2章、第7章和第8章并统稿。限于编者水平，书中疏漏与不足之处在所难免，殷切希望广大读者批评指正。

本书的出版，得到了中国电力出版社的大力支持和协助，谨此表示诚挚的谢意。

编者

2007年12月



目 录

前言

第一版前言

第1章 计算机基础知识	1
1.1 计算机系统概述	1
1.2 计算机中的数制	3
1.3 计算机中的数和字符的表示	9
1.4 机器语言、汇编语言、高级语言	13
习题	15
第2章 80x86 微处理器及系统结构	16
2.1 80x86 微处理器	16
2.2 80x86 微处理器的寄存器	18
2.3 80x86 存储器的组织	23
2.4 80x86 微处理器的工作模式	27
习题	29
第3章 80x86 指令系统	31
3.1 寻址方式	31
3.2 8086 指令系统	35
3.3 80x86 与 Pentium 扩充和增加的指令	64
习题	69
第4章 汇编语言程序格式	72
4.1 汇编语言格式	72
4.2 汇编语句参数	74
4.3 汇编伪指令语句	78
4.4 汇编语言程序的开发	87
4.5 宏指令与条件汇编	96
4.6 结构与记录	101
习题	104
第5章 基本程序设计	108
5.1 顺序程序设计	108
5.2 分支程序设计	113
5.3 循环程序设计	118

5.4	DOS 系统功能调用	133
5.5	子程序设计	136
5.6	具有模块结构的程序设计	147
	习题	150
第 6 章	应用程序设计	152
6.1	算术运算程序设计	152
6.2	非数值处理程序设计	167
	习题	187
第 7 章	输入、输出和中断程序设计	189
7.1	概述	189
7.2	输入、输出程序设计	191
7.3	8086 中断系统	199
7.4	中断处理程序设计	206
7.5	BIOS 功能调用	211
	习题	223
第 8 章	高级语言与汇编语言混合编程	224
8.1	混合编程概述	224
8.2	C 语言程序调用汇编模块	228
8.3	汇编语言与 C 语言的其他协作方式	238
8.4	JAVA 程序调用汇编程序	240
	习题	242
附录 A	8086 指令系统速查表	244
附录 B	汇编出错信息一览	250
附录 C	DOS 系统功能调用 (INT 21H) 表	255
附录 D	ASCII 码字符表	260
	参考文献	262

第1章 计算机基础知识

1.1 计算机系统概述

一个完整的计算机系统是由硬件系统和软件系统两大部分组成的。计算机硬件(Hardware)是构成计算机的各种物质实体的总和。计算机软件(Software)是计算机上全部可运行程序的总和。硬件是软件建立和依托的基础,软件是计算机系统的灵魂。没有软件的计算机称为裸机,不能供用户直接使用;而没有硬件对软件的物质支持,软件的功能则无从谈起。所以应该把计算机系统看作一个整体,它既包括硬件也包括软件,两者不可分割。

1.1.1 硬件

自1946年计算机诞生以来,虽然计算机制造技术已经发生了巨大的变化,但就其体系而言,都基于同一个原理:存储程序和程序控制的原理。其硬件部分都由五大功能部件组成,如图1.1所示。

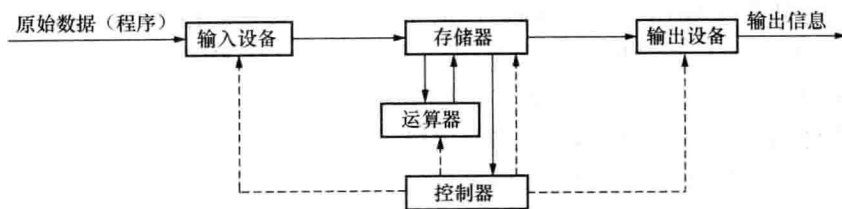


图 1.1 计算机结构示意图

图1.1中实线带箭头“→”代表数据或指令,在机器内部表现为二进制数;虚线带箭头“--->”代表控制信号,在机器内部起控制作用。计算机的工作,正是通过这两种不同类型的信息流动完成的。由图1.1可见,计算机硬件是由输入设备、输出设备、运算器、控制器、存储器五大部件构成的。连接各部件的是总线(BUS),总线是连接计算机内部各部件的一簇公共信号线,是计算机中传送信息的公共通道。它包括地址总线(Address Bus)、数据总线(Data Bus)、控制总线(Control Bus)。

地址总线用来传送地址信息,数据总线用来传送数据信息,控制总线用来传送控制信号。也就是说总线上传送的是地址信息、数据信息、控制信息。

1. 输入设备

计算机要进行数据处理,必须将程序和数据送到内存,转换为计算机能够识别的电信号,这样的设备叫输入设备(Input Unit),如键盘、鼠标和扫描仪等。

2. 输出设备

将主机的信息输出时,就要产生与输出信息相对应的各种电信号,并在显示器上显示、在打印机上打印,或在外存储器上存放等,能将计算机内部的信息传递出来的设备就是输出设备(Output Unit)。

3. 运算器

运算器 (Arithmetic Logical Unit, ALU) 的功能是进行算术运算和逻辑运算。算术运算是指加、减、乘、除等; 逻辑运算泛指非算术运算, 如移位、与、或、非运算等。运算器在控制器的控制下, 从内存中取出数据送到运算器中进行处理, 处理的结果再送回存储器。

4. 控制器

控制器 (Control Unit) 的功能是从内存中依次取出指令, 分析指令并产生相应的控制信号, 送至各个部件, 指挥并协调计算机的各个部件工作。它是统一协调各部件的中枢, 也是“计算机中的计算机”, 它的机理主要是采用内部存储控制信号来实现的。

5. 存储器

存储器 (Memory Unit) 是用来存放程序 and 数据的, 分为主存储器和辅助存储器。

(1) 主存储器 (Main Memory, 简称主存), 又称为内存储器 (简称内存)。在控制器的控制下, 与运算器、输入/输出设备交换信息。目前, 计算机的内存都是采用超大规模的半导体集成器件。它由随机读/写存储器 RAM (Random Access Memory) 和只读存储器 ROM (Read Only Memory) 组成。在 RAM 中的程序和数据, 一旦关机就会全部丢失。主存的速度比运算器的速度慢, 因此在中央处理器内部增加了高速缓冲存储器 (Cache), 以便在速度上和中央处理器相匹配。高速缓冲存储器是利用了程序和数据的局部性原理。

(2) 辅助存储器 (Auxiliary Memory), 也称为外存储器, 简称外存, 是内存的补充和后盾, 当用到外存中的程序和数据时, 才将它们从外存调入内存, 所以外存只同内存交换信息。

外存分为磁表面存储器、光存储器和半导体闪速存储器等三大类。

磁表面存储器分为磁盘存储器和磁带存储器两种, 它们是将磁性材料沉积在盘片 (或带) 的基体上形成记录介质, 并以绕有线圈的磁头与记录介质的相对运动来写入或读出信息。

光存储器主要是光盘 (Optical Disk)。光盘的记录原理不同于磁盘, 它是利用激光束在具有感光特性的表面上存储信息。

闪速存储器 (Flash Memory), 就是通常所说的优盘。自 2000 年闪速存储器被发明和应用以来, 闪速存储器在我国发展很快, 然而作为闪速存储技术中的存储主体——闪速存储器, 其实早在 1980 年就被日本东芝公司申请了专利。多年的发展过程中, 闪速存储器技术经过了多次变革和发展, 但其变化的总体趋势一直都是存储容量越来越大、数据读/写速度越来越快、性能价格比越来越高。

现代计算机将运算器和控制器集成在一起, 称为 CPU (Central Processing Unit, 中央处理器)。而中央处理器和内存又组成了主机。I/O 设备 (Input/Output Unit, 输入/输出设备) 统称为外部设备, 简称外设。

主机通过接口与外设相互连接, 接口是外设与主机进行信息交换的电路。

1.1.2 软件

具有相同硬件的计算机, 配上不同的软件系统, 它们的工作效率会有较大的差别。一台好的计算机不仅需要高档的硬件系统, 还需配有优秀的软件系统。计算机软件系统是指计算机上可运行的全部程序、文档及各种数据的集合。

可见, 计算机软件是计算机系统的重要组成部分, 它可以分成系统软件和应用软件两大类。

系统软件是一组程序, 这些程序是用户使用计算机时, 为产生、准备和执行用户程序所必需的。应用软件则是用户自行编制的各种程序、文档及各种数据的集合。

系统软件的核心称为操作系统（Operating System, OS），是系统程序、文档及各种数据的集合，主要管理计算机系统的全部硬件资源、软件资源及数据资源，控制程序运行，改善人机界面，为其他应用软件提供支持等，使计算机系统所有资源最大限度地发挥作用，为用户提供方便、有效、友善的服务界面。

操作系统通常是最靠近硬件的一层系统软件，它把硬件裸机改造成为功能完善的一台虚拟机，使得计算机系统的使用和管理更加方便、计算机资源的利用效率更高、上层的应用程序可以获得比硬件提供的功能更多的支持。

1.1.3 用户与计算机软件和硬件之间的关系

归纳起来，硬件是计算机系统中看得见的物理实体，而软件则是计算机系统中各种程序的集合。在软件的组成中，系统软件是人与计算机进行信息交换、通信对话、按人的思维对计算机进行控制和管理的工具。

当然，在计算机系统中并没有一条明确的硬件与软件的分界线，软、硬件之间的界限是经常变化的。

1.2 计算机中的数制

1.2.1 计算机中数的表示

计算机中各种信息都采用二进制数的形式来传送、存储和加工。

1. 二进制数的基本概念

二进制（Binary）是“逢二进一”的计数方法，有“0”和“1”两个数码。计算机的机内数据，不论是数值型的（Numeric）还是非数值型（Non-numeric），如数字、文字、图形、图像、声音等信息，都是用二进制数来表示的。在计算机中若干位二进制数表示一个数或者一条指令，前者称为数据字，后者称为指令字。

2. 位

位（bit），也称为比特，是英文 bit（binary digit 的缩写）的译音，常用“bit”或小写“b”表示，是计算机中最小的信息单位，是用 0 或 1 来表示的一个二进制数位。

3. 字节

8 位二进制数为一个字节（Byte），常以大写字母“B”表示。字节是最基本的数据单位。一个字节可放一个 ASCII 码，两个字节可放一个汉字国标码。

4. 字

字（Word）是计算机中信息交换、加工、存储的基本单元。用“W”表示，一个字由一个或者若干个字节构成。字的长度叫字长，是 CPU（中央处理器）内能直接参与运算的二进制位数。

双字（Double Word）是指两个连续字，四个字（Quad Word）是指四个连续字，十字节（Ten Bytes）是指十个连续字节，节（Paragraph）是指十六个连续字节。

1.2.2 计算机采用的数制

日常生活中最熟知的计数进制是十进制，特点是“逢十进一”。

1. 十进制数（D）

十进制数（Decimal）的特点是：

(1) 数码有十个: 0、1、2、3、4、5、6、7、8、9。

(2) 逢 10 进 1, 借 1 当 10。

十进制数按权展开方法是: 设任意一个十进制数 D , 具有 n 位整数, 即 $D_{n-1}D_{n-2}\cdots D_1D_0$ 。十进制数可以表示为

$$D = D_{n-1} \times 10^{n-1} + D_{n-2} \times 10^{n-2} + \cdots + D_1 \times 10^1 + D_0 \times 10^0 \quad (1.1)$$

权是以 10 为底的幂, 将上式称为“按权展开式”。

例如: $(8765)_{10}$ 按权展开为

$$\begin{aligned} (8765)_{10} &= 8 \times 10^3 + 7 \times 10^2 + 6 \times 10^1 + 5 \times 10^0 \\ &= 8000 + 700 + 60 + 5 \end{aligned} \quad (1.2)$$

2. 二进制数 (B)

二进制数的特点是:

(1) 数码有两个: 0、1。

(2) 逢 2 进 1, 借 1 当 2。

二进制数按权展开方法是: 设任意一个二进制数 B , 具有 n 位整数, 即 $B_{n-1}B_{n-2}\cdots B_1B_0$ 。十进制数可以表示为

$$D = B_{n-1} \times 2^{n-1} + B_{n-2} \times 2^{n-2} + \cdots + B_1 \times 2^1 + B_0 \times 2^0 \quad (1.3)$$

权是以 2 为底的幂。

例如: $(111010)_2$ 按权展开为

$$\begin{aligned} (111010)_2 &= 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\ &= (58)_{10} \end{aligned} \quad (1.4)$$

3. 八进制数 (O 或 Q)

八进制数 (Octal) 的特点是:

(1) 数码有八个: 0、1、2、3、4、5、6、7。

(2) 逢 8 进 1, 借 1 当 8。

八进制数按权展开方法是: 设任意一个八进制数 Q , 具有 n 位整数, 即 $Q_{n-1}Q_{n-2}\cdots Q_1Q_0$ 。十进制数可以表示为

$$D = Q_{n-1} \times 8^{n-1} + Q_{n-2} \times 8^{n-2} + \cdots + Q_1 \times 8^1 + Q_0 \times 8^0 \quad (1.5)$$

权是以 8 为底的幂。

例如: $(1765)_8$ 按权展开为

$$(1765)_8 = 1 \times 8^3 + 7 \times 8^2 + 6 \times 8^1 + 5 \times 8^0 = (1013)_{10} \quad (1.6)$$

4. 十六进制数 (H)

十六进制数 (Hexadecimal) 的特点是:

(1) 数码有十六个: 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。

(2) 逢 16 进 1, 借 1 当 16。

其中, 数码 A、B、C、D、E、F 分别代表十进制数 10、11、12、13、14、15。

十六进制数按权展开方法是: 设任意一个十六进制数, 具有 n 位整数, 即 $H_{n-1}H_{n-2}\cdots H_1H_0$ 。十进制数可以表示为

$$D = H_{n-1} \times 16^{n-1} + H_{n-2} \times 16^{n-2} + \cdots + H_1 \times 16^1 + H_0 \times 16^0 \quad (1.7)$$

权是以 16 为底的幂。

例如： $(2E0F)_{16}$ 按权展开为

$$(2E0F)_{16} = 2 \times 16^3 + 14 \times 16^2 + 0 \times 16^1 + 15 \times 16^0 = (11791)_{10} \quad (1.8)$$

二进制数、八进制数、十进制数和十六进制数的对照表，见表 1.1。

在程序设计中，为了区分不同进制的数，通常在数字后用一个英文字母作后缀以示区别。对十进制数，数字后加 D 或不加，如 100D 或 100；对二进制数，数字后加 B，如 101011B；对八进制数，数字后面应该加 O，但是字母 O 和数字 0 在书写时不易区分，为此常常在八进制数的后面加 Q，如 23456Q；对十六进制数，数字后面加 H，如 23ACDH。

表 1.1 各种进制数对照表

十进制	二进制	八进制	十六进制	十进制	二进制	八进制	十六进制
0	0	0	0	8	1000	10	8
1	1	1	1	9	1001	11	9
2	10	2	2	10	1010	12	A
3	11	3	3	11	1011	13	B
4	100	4	4	12	1100	14	C
5	101	5	5	13	1101	15	D
6	110	6	6	14	1110	16	E
7	111	7	7	15	1111	17	F

1.2.3 不同数制的相互转换

1. 十进制数转换为二进制数、八进制数、十六进制数

十进制整数转换为二进制数的方法是除 2 取余；十进制整数转换为八进制数的方法是除 8 取余；十进制整数转换为十六进制数的方法是除 16 取余。

【例 1.1】将十进制数 $(107)_{10}$ 转换为二进制数。

将已知的十进制数的整数部分反复除以 2，直到商是 0 为止，并将每次相除之后所得的余数记录下来。第一次相除之后所得的余数 K_0 为二进制数的最低位，最后一次相除之后所得的余数 K_{n-1} 为二进制数的最高位。 $K_{n-1}K_{n-2} \cdots K_1K_0$ 即为转换所得的数。

转换过程如下：

$$\begin{array}{rcl}
 2 \overline{) 107} & \cdots \cdots \text{余数 } 1 (K_0) & \text{(低位)} \\
 2 \overline{) 53} & \cdots \cdots \text{余数 } 1 (K_1) & \\
 2 \overline{) 26} & \cdots \cdots \text{余数 } 0 (K_2) & \\
 2 \overline{) 13} & \cdots \cdots \text{余数 } 1 (K_3) & \\
 2 \overline{) 6} & \cdots \cdots \text{余数 } 0 (K_4) & \\
 2 \overline{) 3} & \cdots \cdots \text{余数 } 1 (K_5) & \\
 2 \overline{) 1} & \cdots \cdots \text{余数 } 1 (K_6) & \text{(高位)} \\
 0 & &
 \end{array} \quad (1.9)$$

所以 $(107)_{10} = (1101011)_2$

【例 1.2】将十进制数 $(117)_{10}$ 转换为八进制数。

将已知的十进制数的整数部分反复除以 8，直到商是 0 为止，并将每次相除之后所得的余数记录下来。第一次相除之后所得的余数 K_0 为八进制数的最低位，最后一次相除之后所得

的余数 K_{n-1} 为八进制数的最高位。 $K_{n-1}K_{n-2}\cdots K_1K_0$ 即为转换所得的数。

转换过程如下:

$$\begin{array}{rcl}
 8 \overline{) 117} & \cdots \cdots \text{余数 } 5 \ (K_0) & \text{(低位)} \\
 \underline{8 14} & \cdots \cdots \text{余数 } 6 \ (K_1) & \uparrow \\
 8 \overline{) 14} & \cdots \cdots \text{余数 } 1 \ (K_2) & \text{(高位)} \\
 \underline{8 1} & & \\
 0 & &
 \end{array} \quad (1.10)$$

所以 $(117)_{10} = (165)_8$

【例 1.3】 将十进制数 $(687)_{10}$ 转换为十六进制数。

将已知的十进制数的整数部分反复除以 16, 直到商是 0 为止, 并将每次相除之后所得的余数记录下来。第一次相除之后所得的余数 K_0 为十六进制数的最低位, 最后一次相除之后所得的余数 K_{n-1} 为十六进制数的最高位。 $K_{n-1}K_{n-2}\cdots K_1K_0$ 即为转换所得的数。

转换过程如下:

$$\begin{array}{rcl}
 16 \overline{) 687} & \cdots \cdots \text{余数 } 15 \ (K_0) & \text{(低位)} \\
 \underline{16 42} & \cdots \cdots \text{余数 } 10 \ (K_1) & \uparrow \\
 16 \overline{) 42} & \cdots \cdots \text{余数 } 2 \ (K_2) & \text{(高位)} \\
 \underline{16 2} & & \\
 0 & &
 \end{array} \quad (1.11)$$

所以 $(687)_{10} = (2AF)_{16}$

2. 二进制数、八进制数、十六进制数转换为十进制数

(1) 二进制数转换为十进制数的方法是: 将二进制数 (基数为 2) 按权展开相加, 即可得到相应的十进制数。

【例 1.4】 将二进制数 $(1101)_2$ 转换为十进制数。

$$\begin{aligned}
 (1101)_2 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
 &= (13)_{10}
 \end{aligned} \quad (1.12)$$

(2) 八进制数、十六进制数转换为十进制数方法是: 将八进制数 (基数为 8)、十六进制数 (基数为 16) 按权展开相加, 即可得到相应的十进制数。

3. 二进制数和八进制数的相互转换

因为二进制的进位基数是 2, 八进制的进位基数是 8, 所以三位二进制数对应一位八进制数。所以, 二进制数转换为八进制数的方法是: 从右向左, 三位一组, 最高位不足三位时, 左边添 0 补足三位, 然后将每组的三位二进制数用相应的八进制数表示, 即可得到对应的八进制数。

反之, 将八进制数转换为二进制数时, 每一位八进制数用对应的三位二进制数表示即可。

【例 1.5】 将二进制数 $(1011101)_2$ 转换为八进制数。

$$\begin{array}{ccc}
 001 & 011 & 101 \\
 1 & 3 & 5
 \end{array} \quad (1.13)$$

所以 $(1011101)_2 = (135)_8$

4. 二进制数和十六进制数的相互转换

因为二进制的进位基数是 2, 十六进制的进位基数是 16, 所以四位二进制数对应一位十六进制数。

所以,二进制数转换为十六进制数的方法是:从右向左,四位一组,最高位不足四位时,左边添0补足四位,然后将每组的四位二进制数用相应的十六进制数表示,即可得到对应的十六进制数。

反之,将十六进制数转换为二进制数时,每一位十六进制数用对应的四位二进制数表示即可。

【例 1.6】 将二进制数 $(11001011101)_2$ 转换为十六进制数。

转换过程如下:

$$\begin{array}{ccc} 0110 & 0101 & 1101 \\ 6 & 5 & D \end{array} \quad (1.14)$$

所以 $(11001011101)_2 = (65D)_{16}$

由以上讨论可知,二进制数与八进制数、十六进制数之间的转换比较容易、直观。所以在程序设计中,通常将书写起来很长且容易出错的二进制数用简捷的八进制数或十六进制数表示。

1.2.4 二进制数的算术运算

1. 加法运算

二进制加法运算规则是逢2进1,即

$$0+0=0 \quad 0+1=1 \quad 1+0=1 \quad 1+1=10$$

【例 1.7】 $1101+1011=11000$ (1.15)

2. 减法运算

二进制减法运算规则是借1当2,即

$$0-0=0 \quad 1-0=1 \quad 1-1=0 \quad 0-1=1 \text{ (向高位借1)}$$

【例 1.8】 $1101-1011=0010$ (1.16)

3. 乘法运算

二进制乘法运算规则是0乘以任何数得0,1乘以任何数得该数,即

$$0 \times 0=0 \quad 0 \times 1=0 \quad 1 \times 0=0 \quad 1 \times 1=1$$

【例 1.9】 $1101 \times 1011=10001111$ (1.17)

4. 除法运算

二进制除法运算规则是0除以1得0,1除以1得1,0做除数无意义,即

$$0 \div 1=0 \quad 1 \div 1=1$$

【例 1.10】 $10001111 \div 1011=1101$ (1.18)

1.2.5 计算机中的逻辑运算

如果给一个变量赋以逻辑属性,这个变量就为逻辑变量。逻辑变量的值只有两个,即“真”与“假”(“T”与“F”),或“是”与“否”(“Y”与“N”)。用二进制数的“1”和“0”来表示这种逻辑值就十分方便了,一般情况下,“1”代表“T”,“0”代表“F”。

逻辑运算包括逻辑非、逻辑或、逻辑与这三种基本运算,或由这三种基本运算复合而成的如与非、或非、异或等的逻辑组合。

在非数值运算领域中,经常使用逻辑运算,例如利用逻辑运算进行两数的比较,或者从某个数中选取某几位等操作。再比如当利用计算机进行过程控制时,可以利用逻辑运算对一组输入的开关量做出判断,以确定哪些开关是闭合的,哪些开关是断开的。总之,在非数值应

用的广大领域中, 逻辑运算是非常有用的。

计算机中的逻辑运算, 主要是指逻辑乘、逻辑加 (或)、逻辑非、逻辑异或等四种基本运算。

1. 与运算 (逻辑乘, Logic Multiplication)

当两个条件同时为真时, 结果才为真; 当两个条件中任意一个为假, 结果必为假。这种逻辑关系称为“与”逻辑。通常用符号 $\times$ 、 $\wedge$ 、 $\cdot$ 、 $\cap$ 或 AND 等来表示“与”。与运算的规则是

$$0 \times 0 = 0 \quad 0 \times 1 = 0 \quad 1 \times 0 = 0 \quad 1 \times 1 = 1$$

若两个逻辑数 X 、 Y 分别为

$$X = X_n X_{n-1} \cdots X_1 X_0, \quad Y = Y_n Y_{n-1} \cdots Y_1 Y_0$$

则

$$X \wedge Y = Z = Z_n Z_{n-1} \cdots Z_1 Z_0 \quad (1.19)$$

其中, $Z_i = X_i \wedge Y_i \quad (i=0, 1, 2, \cdots, n)$

【例 1.11】 $X=100101$, $Y=101101$, 求 $X \wedge Y$ 。

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

即 $X \wedge Y = 100101$

2. 或运算 (逻辑加, Logic Addition)

当两个条件中任意一个为真, 结果就为真; 当两个条件同时为假时, 结果才为假。这种逻辑关系称为“或”逻辑。通常用符号 $+$ 、 $\vee$ 、 $\cup$ 或 OR 等来表示“或”。或运算的规则是

$$0 + 0 = 0 \quad 0 + 1 = 1 \quad 1 + 0 = 1 \quad 1 + 1 = 1$$

设两个数 X 、 Y 分别为

$$X = X_n X_{n-1} \cdots X_1 X_0, \quad Y = Y_n Y_{n-1} \cdots Y_1 Y_0$$

对 X 、 Y 求逻辑加, 则

$$X \vee Y = Z = Z_n Z_{n-1} \cdots Z_1 Z_0 \quad (1.20)$$

其中, $Z_i = X_i \vee Y_i \quad (i=0, 1, 2, \cdots, n)$

【例 1.12】 $X=100101$, $Y=101101$, 求 $X \vee Y$ 。

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

即 $X \vee Y = 101101$

3. 非运算 (Logic Negation)

非运算是进行“求反”运算, 又称否运算。通常在逻辑变量上方加一横线, 如 $\bar{B}$ 。也可用符号“ $\neg$ ”来表示, 如 $\neg B$ 。非运算的规则是

如 $B=1$, 则 $\neg B=0$; 如 $B=0$, 则 $\neg B=1$ 。

【例 1.13】 设 $X=10101010$

则 $\neg X=01010101$

4. 异或逻辑

对两个数进行异或操作就是按位求它们的模 2 和, 故异或逻辑又有“按位加”之称, 常用记号“ $\oplus$ ”来表示。

若两个数 X 、 Y 分别为

$$X = X_n X_{n-1} \cdots X_1 X_0, \quad Y = Y_n Y_{n-1} \cdots Y_1 Y_0$$

则

$$X \oplus Y = Z = Z_n Z_{n-1} \cdots Z_1 Z_0 \quad (1.21)$$

其中 $Z_i = X_i \oplus Y_i \quad (i=0, 1, 2, \cdots, n)$

【例 1.14】 $X=100101$, $Y=101101$, 求 $X \oplus Y$ 。

$$\begin{array}{r} X \quad \quad 100101 \\ Y \quad \quad 101101 \\ \hline \quad \quad 001000 \end{array}$$

即 $X \oplus Y = 001000$

1.3 计算机中的数和字符的表示

字符是计算机中使用最多的信息之一, 是人与计算机通信、交互作用的重要媒介。在计算机中, 为每个字符指定一个确定的编码, 作为识别与使用这些字符的依据。这些编码的值是用一定位数的二进制码表示的。

1.3.1 字符表示

目前, 国际上使用的字母、数字和符号的信息编码系统是采用美国国家信息交换标准字符码 (American Standard Code for Information Interchange), 简称为 ASCII 码。它有 7 位码版本和 8 位码版本两种。国际上通用的 ASCII 码是 7 位码 (即用 7 位二进制数表示一个字符), 总共有 128 个字符 ($2^7=128$), 其中包括 26 个大写英文字母, 26 个小写英文字母, 0~9 共 10 个数字, 34 个通用控制字符 (NUL~SP 和 DEL) 和 32 个专用字符 (标点符号和运算符)。具体编码表见附录 D。

用一个字节 (8 位二进制数) 表示 7 位 ASCII 码时, 最高位为 0。它的范围为 00000000B~01111111B。8 位 ASCII 码称为扩充 ASCII 码, 是 8 位二进制字符编码, 它的范围为 00000000B~11111111B, 其中 00000000B~01111111B 为基本部分, 范围为 0~127, 这 128 种最高位为 0; 10000000B~11111111B 为扩充部分, 范围为 128~255, 也有 128 种, 其最高位为 1。因此 8 位 ASCII 码可以表示 256 种不同的字符。尽管对扩充部分的 ASCII 码美国国家标准信息协会已给出定义, 但在实际中多数国家将 ASCII 码扩充部分规定为自己国家语言的字符代码, 如中国把扩充 ASCII 码作为汉字的机内码。在我国国家标准 GB 2312—1980《信息交换用汉字编码字符集 基本集》中还规定使用两个字节 (每个字节的最高位置 0) 对应一个汉字进行编码, 称为国标码, 而把每个字节的最高位都置 1, 作为对应的汉字的机内码 (也称汉字的 ASCII 码)。

1.3.2 BCD 码

虽然二进制运算规则简单、计算机处理容易实现,但二进制数不直观、书写容易出错,加之人们习惯用十进制数形式,因此计算机中的数字有时也用十进制形式表示。这里介绍常用的表示十进制数的 BCD 编码。

BCD (Binary Code Decimal) 码叫做二进制编码的十进制数,也称二-十进制数。一位十进制数可以用 4 位二进制数表示,其表示方法有多种,通常用的是 8421BCD 码,即 4 位二进制数自左至右其权分别为 8、4、2、1。用二进制数的 0000~1001 分别表示 8421BCD 码的 0~9。

BCD 码有压缩 BCD 码和非压缩 BCD 码两种形式。

1. 压缩 BCD 码

每位压缩 BCD 码占用 4 个二进制位,一个字节 (8 位) 可以存放两位压缩 BCD 码。

例如要把十进制数 56 转换成压缩 BCD 码。则有

$$(56)_{10} = (01010110)_{\text{压缩 BCD}} \quad (1.22)$$

2. 非压缩 BCD 码

每位非压缩 BCD 码占用一个字节,其中高 4 位为 0,低 4 位为 BCD 码。

1.3.3 无符号数和带符号数

1. 无符号数

所谓无符号数 (Unsigned Number),就是整个机器字长的全部二进制位均表示数值位,也就是没有符号位。相当于数的绝对值,例如:

$X_1 = 01001$ 表示无符号数 9。

$X_2 = 11001$ 表示无符号数 25。

机器字长为 n 位的无符号数的表示范围是 $0 \sim (2^n - 1)$,此时二进制的最高位也是数值位,其权值等于 2^n 。若字长为 8 位,则数的表示范围为 $0 \sim 255$ 。

一般计算机中都设置有一些无符号数的运算和处理指令。如 Intel 8086 中的 MUL 和 DIV 指令就是无符号数的乘法和除法指令,还有一些条件转移指令也是专门针对无符号数的。

2. 带符号数

然而,大量用到的数据还是带符号数 (Signed Number),即正、负数。在日常生活中用 “+” “-” 号加绝对值来表示数值的大小,用这种形式表示的数值在计算机技术中称为 “真值”。

对于数的符号 “+” 或 “-”,计算机是无法识别的,因此需要把数的符号数码化。通常,约定二进制数的最高位为符号位,“0” 表示正号,“1” 表示负号。这种在计算机中使用的表示数的形式称为机器数,常见的机器数有原码、反码、补码等三种不同的表示形式。

带符号数的最高位被用来表示符号位,而不再表示数值位。前例中的 X_1 和 X_2 在这里的含义变为

$X_1 = 01001$ 表示有符号数 +9。

$X_2 = 11001$ 根据机器数的不同形式表示不同的值,如果是原码,则表示 -9,补码则表示 -7,反码则表示 -6。

3. 数的补码表示

计算机中的数是用二进制来表示,数的符号也是用二进制数表示。在机器中,把一个数