

942254

TP36
1243

高等学校教材

微型计算机原理

山东工业大学 张荣祥 主编



TP36
1243

942254

TP36
1243

高等学校教材

微型计算机原理

山东工业大学 张荣祥 主编

水利电力出版社

高等学校教材
微型计算机原理
山东工业大学 张荣祥 主编

*

水利电力出版社出版

(北京三里河路6号)

新华书店北京发行所发行·各地新华书店经售

北京市地质局印刷厂印刷

*

787×1092毫米 16开本 20印张 448千字
1992年6月第一版 1992年6月北京第一次印刷
印数 0001—2660册

ISBN 7-120-01539-7/TP·54

定价 5.20元

(京)新登字 115 号

内 容 简 介

本书以 Intel 公司的 8086 微处理机为背景,对微型计算机的硬件、软件、系统设计等方面做了系统的阐述和分析,使读者能由此掌握微型计算机的一般原理和基本应用知识。

全书共分七章:概论;微处理机及存储器结构;8086 指令系统;8086 汇编语言及程序设计方法;中断技术;输入/输出接口技术;系统的组成和设计。书中各章后都附有习题,书后的附录中系统地列出了 8086 指令系统,并给出了个人计算机的逻辑框图。

本书是全国高等院校电力工程类专业的统编教材,既适合作高等院校有关专业及培训班的教材,亦可供计算机专业人员和从事微型计算机应用开发的工程技术人员参考。

前 言

电子计算机尤其是微机技术的迅速发展,使设计、制造、试验等生产系统的各个方面都已广泛使用微型计算机,它已成为技术人员必须掌握的工具。在高等院校中亦越来越重视学生对于微机原理的学习,以及培养他们应用微机解决各类专业的实际问题的能力。这些问题不仅是复杂的科学计算,而且越来越多的是实时检测和控制问题以及范围广泛的信息处理问题,因此高等院校中很多专业都开设了微机原理及其应用的课程。近几年来,相应的教材已经出了不少,但绝大多数教材的内容均以 8 位微机为基础,尤其是以 Z-80 为 CPU 的微机原理作为基本内容。然而,近年来 16 位微处理器及其相应的微机系统在国内已开始大量使用,CPU 芯片和单片微机的价格不断下降。对于工程技术专业,所遇到的实际问题常常需要在复杂运算或控制的同时,满足快速处理、高精度和高可靠性的要求,16 位微处理器具有 8 位微处理器所没有的独特的优点和适应性。考虑到微机技术的飞速发展和 90 年代专业技术的需要,本书以 Intel 8086 为 CPU 的微机系统的原理及应用作为基本内容。学生学习这些基本内容并通过一定数量的上机操作实验后将不难掌握 8 位微处理器及其他类型 16 位微处理器的原理和应用,同时也为学习在 90 年代将成为主流的 32 位微机系统打好基础。

本书是全国高等院校电力工程类专业的统编教材。它是按能源部高等学校电力工程类专业教学委员会自动远动组在 1989 年 4 月杭州会议上审订的编写大纲编写而成,符合《微型计算机原理》课程 70 学时的要求,在内容上力求贯彻“少而精”的原则。

《微型计算机原理》课程属于专业基础课的范畴,课程中的原理性内容为应用性内容打基础,而它们又都为今后的专业领域内实际应用微机打基础。这类应用主要面向检测、控制和信息处理,因此,本书内容是从应用的角度来取舍的,论述了微机的基本概念、基本功能原理及基本应用方法,并且注意到了使学生兼有硬件和软件两方面的知识。

参加本书编写的同志有:山东工业大学张荣祥教授,王侠副教授,包成谋高级工程师。全书由张荣祥主编,包成谋还对书中所有汇编程序举例作了校核。本书主审人浙江大学韩桢祥教授对全书进行了认真仔细的审阅,并提出了许多宝贵意见;山东工业大学电力系计算中心孙怡同志承担了大部分书稿的计算机打印和制图工作;高等学校电力工程类专业教学委员会自动远动组的委员们对本书的编写提出了不少很好的建议和意见,谨在此一并表示衷心的感谢。

限于编者水平,错误和不当之处在所难免,切望读者批评指正。

编 者

1991 年 3 月

2006/08

目 录

前 言

第一章 概论	1
第一节 微型计算机系统的基本结构	1
第二节 计算机中的数制和码制	3
一、数和进位计数制	3
二、不同进位制数之间的转换	5
三、十进制数的二进制编码及 ASCII 码	8
第三节 机器字的概念和带符号数的表示法	10
一、8086 的三种数据类型	10
二、机器字的概念	12
三、带符号数的表示法	12
第四节 微型计算机的发展和应用	15
习题	17
第二章 微处理器及存储器结构	19
第一节 概述	19
第二节 8086 CPU 结构	20
一、执行单元 EU	20
二、总线接口单元 BIU	20
三、寄存器组	22
第三节 存储器结构	24
一、存储器的分段技术	24
二、实际地址的形成	25
三、字节和字的传送	26
第四节 堆栈	28
第五节 输入和输出	28
第六节 8086 CPU 的引脚及其功能	29
一、8086 总线周期	30
二、8086 的引脚定义	31
第七节 8086 的最小模式和最大模式系统	37
习题	39
第三章 8086 的寻址方式和指令系统	40
第一节 机器语言的指令格式和编码	40
一、加法指令中各个字节的分析	40
二、存储器有效地址的计算方式	44
三、举例	44
四、段寄存器的隐含访问和段超越前缀	44

第二节 8086 寻址方式	45
一、立即数寻址	45
二、寄存器寻址	45
三、存储器寻址	45
第三节 8086 指令系统	48
一、数据传送指令	48
二、算术运算指令	53
三、逻辑运算指令	62
四、字符串操作指令	67
五、转移指令	72
六、控制性指令	76
七、中断指令	78
八、8086 指令系统总表简介	79
习题	79
第四章 8086 汇编语言及程序设计方法	82
第一节 8086 汇编语言简介	82
一、机器语言、汇编语言和高级语言	82
二、8086 汇编语言的特点	86
三、8086 汇编语言程序开发	87
第二节 ASM-86 汇编语言程序	90
一、源程序的构成特点	90
二、汇编语言的基本元素	92
三、汇编语言语句的组成	94
四、程序分段和段定义语句	95
五、数据和数据定义语句	103
六、表达式	115
七、过程定义和调用语句	117
第三节 程序设计的基本方法	118
一、程序设计过程	118
二、汇编语言程序的基本形式	119
第四节 实用程序举例	134
一、基本运算程序	134
二、数据处理程序	145
三、用于实时处理的基本程序	150
第五节 宏汇编语言及程序设计	160
一、概述	160
二、简单的宏定义及调用	161
三、宏定义库和调用	168
习题	174
第五章 中断技术	176
第一节 中断的一般概念	176
第二节 中断类型和中断地址指针表	177

一、中断类型	177
二、中断地址指针表	177
第三节 中断指令和中断的屏蔽	178
一、中断指令	178
二、中断的屏蔽	179
第四节 中断功能	180
一、内部中断功能	180
二、软件中断	181
三、非屏蔽中断	182
四、外部硬件中断	182
第五节 8259A 可编程中断控制器	186
一、8259A 的方块图	186
二、8259A 的内部结构	187
三、8259A 的编程	188
四、主从结构中中断接口	194
习题	196
第六章 输入/输出接口技术	197
第一节 概述	197
第二节 I/O 地址空间和数据传送	198
第三节 输入/输出指令和总线周期	199
一、输入/输出指令	199
二、输入和输出总线周期	201
第四节 可编程并行 I/O 接口	201
一、可编程并行 I/O 接口 8255A 的基本结构和功能	201
二、8255A 的工作方式	203
三、用 8255A 实现并行 I/O 端口	209
第五节 存储器映象 I/O	209
第六节 8255A 与 A/D 和 D/A 转换器的连接	211
第七节 可编程串行 I/O 接口	213
一、概述	213
二、同步和异步通信	215
三、串行通信的差错控制	225
四、8251A 可编程通信接口	230
五、HDLC 接口	238
习题	240
第七章 系统的组成和设计	242
第一节 地址锁存和 8282 锁存器的使用	242
第二节 数据功率放大和 8286 收发器的使用	245
第三节 8254 可编程定时器/计时器的使用	247
第四节 只读存储器和随机存取存储器与 CPU 的连接	252
第五节 直接存储器存取和 8257 DMA 控制器的使用	257
一、概述	257

二、8257 DMA 控制器	258
第六节 多处理机系统的结构和协调	266
一、队列状态和总线封锁功能	268
二、8086 的多道处理系统	270
三、微型计算机通信网	286
习题	287
附 录	288
附录一 ASCII 码	288
附录二 8086 指令系统表	289
附录三 RS-232-C 连接器标准	304
附录四 浮点记数法	305
附录五 个人计算机逻辑框图及扩充插槽信号	306
参考文献	309

第一章 概 论

第一节 微型计算机系统的基本结构

微型计算机系统和所有其它计算机系统一样是由硬件和软件两大部分组成的。硬件是系统中由各种实在的器件构成的物理装置，它是计算机系统的物质基础。软件则是在硬件上记录和运行的各式各样的程序及其有关资料，软件是使计算机取得实际效能不可缺少的部分。但是从原理上看，计算机的硬件和软件之间常常可以转化，由软件完成的功能可以由硬件来实现，由硬件实现的操作也可以由软件来模拟。例如，乘法操作可以通过子程序来实现，也可以直接由硬件来完成。因此，具有相同功能的计算机系统，其软、硬件功能的分配是可以在一定范围内转化的，主要取决于现有硬件状况下的性能价格比。一般来说，提高软件功能的比例，可以降低造价、提高灵活性和适应性，但执行速度下降，所需存储容量也要增加。

计算机系统的硬件可以分为四个基本功能块，即中央处理单元、存储单元、输入单元和输出单元，如图 1-1 所示。

中央处理单元 CPU (Central Processing Unit) 或称微处理器单元 MPU (Micro Processing Unit) 是系统的核心，它控制整个系统的操作，并在程序的启动下完成所有的运算和逻辑操作。它包括运算器和控制器，两者不论在逻辑关系上或者在结构工艺上都有十分紧密的关系。运算器是一个用于信息加工的部件，它用来对二进制的数据进行算术运算和逻辑运算。控制器的任务则是控制计算机对程序的执行，即按一定的顺序控制程序中的每一条机器指令的执行。

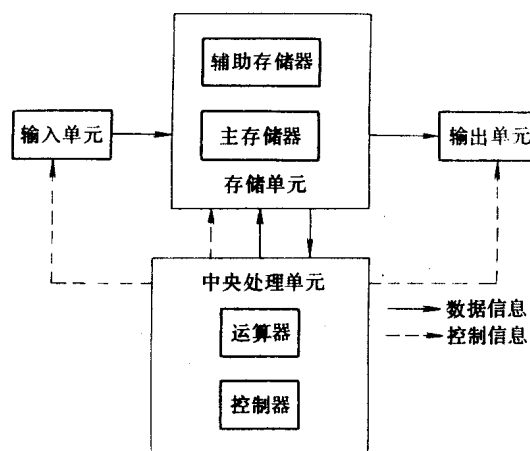


图 1-1 计算机系统的基本功能块图

输入和输出单元是实现人与计算机之间联系的设备。通过输入单元把原始数据或命令及计算程序送入机器中的存储单元，例如程序员从键盘输入一个新的程序，而计算机运算的某些中间结果和最终结果等经处理后得到的信息，又必须通过输出单元送出，例如由打印机打印或在 CRT (Cathode-Ray Tube) 显示器的屏幕上显示出来，人才能看到。在某些简单的应用场合，输入单元只用几个开关、输出单元只用几只发光二极管来实现。由于在计算机内部的一切信息都是以二值代码表示的，与人表达信息的习惯很不相同，因此在输入/输出过程中一般要进行信息变换。

存储单元是用来存放程序和数据的，它能够保持这些信息以便稍后一个时间再作处理

或输出。它分为主存储器(或称内存储器)和辅助存储器(或称外存储器)。存储器中含有大量的存储单元,每个存储单元可以存放一个字节。一个字节可以是一条机器指令,称为指令字;也可以是一个数据,称为数据字。每个存储单元有一个固定的编号,这个编号称为地址。

对存储器最主要的要求是存储容量要大,存取速度要高。一般在计算机系统中采用容量不太大而速度高的内存储器和容量很大而速度较低的外存储器配合进行工作,外存用于长期存储目前不用的信息。在大多数计算机系统中,这个部分常常用存储于磁性媒体上的方法来实现,如磁带、磁盘和磁鼓。内存是较小的存储区,它采用电子存储器件而使内部存储操作比外部存储操作快。在大多数现代计算机中,内存包括只读存储器 (ROM-Read Only Memory) 和随机存取存储器 (RAM-Random Access Memory)。例如在一个游戏用计算机系统中,决定游戏如何玩的程序存储于 ROM 而应用数据则存放在 RAM 中。

由于 CPU 和内存是信息加工处理的主要部分,所以合称为“主机”,而输入/输出单元及外存合称为“外围设备”(或称“外部设备”)。凡是同主机交换信息、完成输入/输出功能的装置都可以定义为外围设备。

在计算机中各部件间信息传送的方式可以用专线式信息传送或总线式信息传送。前者指各部件根据需要分别设置通往其它部件的信息传输线;而后者信息的传送是通过一组总线进行的。在图 1-2(a)中是以一组总线使主机与整个系统的外围设备相连接的,这样可以减少连线,使机器得到简化,易于实现模块化,但总线必须分时使用,使传输速度受到限制,控制也较复杂。

在图 1-2 (a) 的连接方式中,设备和设备控制器通过外围设备接口 (或称 I/O 接口) 同总线相连。设备控制器用来对设备进行启停控制和辅助操作的控制,以及用来指示设备的工作状态等。各种不同设备的设备控制器可以独立于设备之外,也可以包含在设备之中。外围

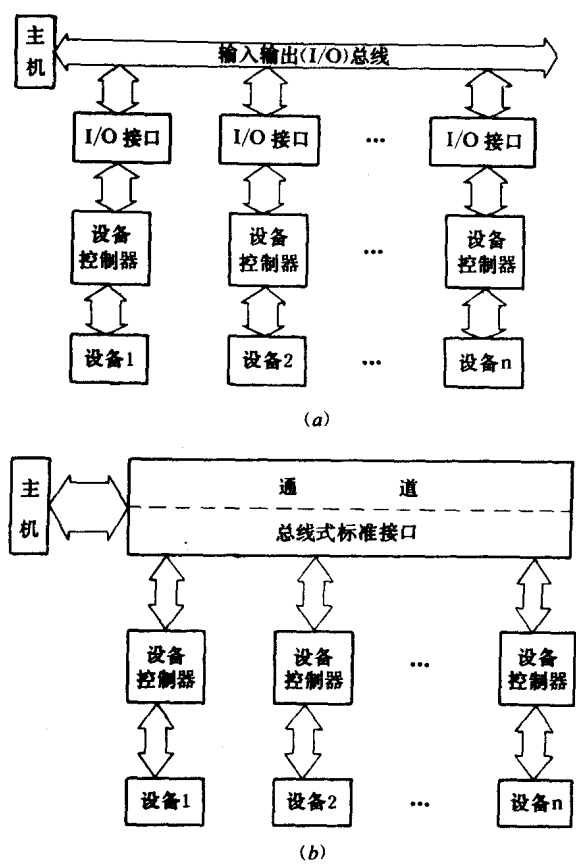


图 1-2 外围设备与主机的连接方式
(a) 总线式; (b) 通道式

设备接口的功能是:实现主机与外围设备之间的通信联络控制;实现数据缓冲以达到主机同外围设备之间的速度匹配;进行简单的数据格式变换等等。此外,接口使外围设备的连接标准化,即配备接口的任何设备都能通过总线同主机相连。在这种连接方式中,信息交换的过程是在主机 (CPU) 控制下进行的。

计算机系统是硬件和软件密切结合的统一整体，除了硬件以外，用户还要通过软件来使用和发挥它的功能。

系统软件主要包括如下部分:

(2) 各种程序设计语言及其编译系统, 可分为面向机器的汇编语言及其汇编程序和面向过程或问题的高级语言及其编译系统两大类。

(3) 调试程序及故障检查和诊断程序。当新编制的程序上机调试时, 调试程序能使用户发现和寻找错误所在。

式中，每一个数位都有一个基值与之相对应，这个基值称为权。小数点左边各数位的权是 10 的正次幂，小数点右边各数位的权是 10 的负次幂。所以这个数可写成

$$9999.99 = 9 * 10^3 + 9 * 10^2 + 9 * 10^1 + 9 * 10^0 + 9 * 10^{-1} + 9 * 10^{-2}$$

这里 * 为计算机中乘号的符号。

每一个数位上的值是该位的数码与权的乘积。于是，任意一个十进制数 D ，可以表示为

$$\begin{aligned} D &= D_{n-1} * 10^{n-1} + D_{n-2} * 10^{n-2} + \dots + D_0 * 10^0 + D_{-1} * 10^{-1} \\ &\quad + \dots + D_{-m} * 10^{-m} \\ &= \sum_{i=-m}^{n-1} D_i * 10^i \end{aligned}$$

式中， n 为整数部分的位数， m 为小数部分的位数； D_i 的值为 0、1……8、9 十个数码中任一个，它取决于一个具体的数。

(二) 二进制数

二进制数的两个基本特点：

- (1) 具有两个不同的数码符号：0 和 1；
- (2) 计数进位原则：逢二进一（减法为借一当二）。

与十进制数相似，同一个数码在不同的数位代表不同的数值，即每一个数位有一个权与之相对应。

一个二进制数的值也可以用它的按权展开式来表示，例如

$$\begin{aligned} (11011.101)_2 &= 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 + 1 * 2^{-1} \\ &\quad + 0 * 2^{-2} + 1 * 2^{-3} \\ &= (27.625)_{10} \end{aligned}$$

式中，小数点左边各数位的权是 2 的正次幂，小数点右边各数位的权是 2 的负次幂。于是，任意一个二进制数 B 可以表示为

$$\begin{aligned} B &= B_{n-1} * 2^{n-1} + B_{n-2} * 2^{n-2} + \dots + B_0 * 2^0 + B_{-1} * 2^{-1} + \dots + B_{-m} * 2^{-m} \\ &= \sum_{i=-m}^{n-1} B_i * 2^i \end{aligned}$$

式中， n 为整数部分的位数， m 为小数部分的位数； B_i 值为 0 或 1，取决于一个具体的数。

(三) 十六进制数

十六进制数的两个基本特点：

- (1) 具有 16 个不同的数码符号：0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F；
- (2) 进位计数原则：逢十六进一（减法为借一当十六）。

同样，同一数码在不同的数位代表不同的值。例如

$$(6C6.11)_{16} = 6 * 16^2 + 12 * 16^1 + 6 * 16^0 + 1 * 16^{-1} + 1 * 16^{-2} = (1734.0664)_{10}$$

任意一个十六进制数 H ，可表示为

$$\begin{aligned}
 H &= H_{n-1} * 16^{n-1} + H_{n-2} * 16^{n-2} + \cdots + H_0 * 16^0 + H_{-1} * 16^{-1} \\
 &\quad + \cdots + H_{-m} * 16^{-m} \\
 &= \sum_{i=-m}^{n-1} H_i * 16^i
 \end{aligned}$$

式中, n 为整数部分的位数, m 为小数部分的位数; H_i 值是 $0 \sim 9$ 和 $A \sim F$ 中任一个数码, 取决于一个具体的数。

二、不同进位制数之间的转换

十进制数、二进制数和十六进制数之间的数值转换对照如表 1-1 所示。

表 1-1 十进制数、二进制数和十六进制数对照表

十进制	二进制	十六进制	十进制	二进制	十六进制	十进制	二进制	十六进制
0	0000	0	6	0110	6	12	1100	C
1	0001	1	7	0111	7	13	1101	D
2	0010	2	8	1000	8	14	1110	E
3	0011	3	9	1001	9	15	1111	F
4	0100	4	10	1010	A	16	10000	10
5	0101	5	11	1011	B	17	10001	11

十进制数是人们在日常生活中用来计数的一种进位计数制。二进制数只有“0”和“1”两个数码, 在数字电路中可用高电平代表“1”, 用低电平代表“0”, 也就是用两种相反的物理状态来表示“1”和“0”。具有两个不同的稳定状态且能相互转换的物理器件很多, 并且二进制的运算规律又较简单, 所以在计算机中采用二进制。但是人们用二进制数表示数值时, 数位很多, 书写不便, 因此为了简便起见, 也广泛采用十六进制数, 就是用一位十六进制数来表示四位二进制数。

(一) 二进制数与十六进制数之间的转换

因为 $2^4 = 16$, 故一位十六进制数相当于 4 位二进制数, 其对应关系见表 1-1。只要熟记它们之间的对应关系, 二进制数与十六进制数之间的相互转换是很简单的。

(1) 二进制数转换成十六进制数: 以小数点为界, 分别向左向右, 每 4 位为一组, 最后不足 4 位者, 前面 (或后面) 补零, 每 4 位二进制数用一位十六进制数代替。

【例 1-1】 $(101100.101)_2 = (? . ?)_{16}$

解:

0010	1100	.	1010
2	C	.	A

$(101100.101)_2 = (2C.A)_{16}$

(2) 十六进制数转换成二进制数: 不论是整数部分还是小数部分, 只要把每一位十六进制数用相应的 4 位二进制数来代替即可。

【例 1-2】 $(4A.B3)_{16} = (? . ?)_2$

解:

4	A	.	B	3
0100	1010	.	1011	0011

$$(4A.B3)_{16} = (1001010.10110011)_2$$

在书写时,为了区分是哪种进位计数制,也可以采用后缀字母以示其进位计数制。用后缀字母 B(Binary)表示该数为二进制数;H(Hexadecimal)表示该数为十六进制数;后缀 D(Decimal)表示该数为十进制数,但一般书写十进制数不加后缀 D。

(二)二进制数或十六进制数转换成十进制数

这种转换很简单,只要将二进制数或十六进制数的按权展开式相加,就得出相对应的十进制数。

在任意一个二进制数 B 的按权展开式中,任一数位的数码为 1,则该位的权为有效权值;若数码为 0,则该位的权为无效权值。所以,只要将二进制数或十六进制数的所有“有效权值”相加,即得对应的十进制数。

二进制数各数位的权值如表 1-2 所示。

表 1-2

2 的正、负次幂

2 的正次幂				2 的负次幂	
$2^0=1$					
$2^1=2$	$2^5=32$	$2^9=512$	$2^{13}=8192$	$2^{-1}=0.5$	$2^{-5}=0.03125$
$2^2=4$	$2^6=64$	$2^{10}=1024$	$2^{14}=16384$	$2^{-2}=0.25$	$2^{-6}=0.015625$
$2^3=8$	$2^7=128$	$2^{11}=2048$	$2^{15}=32768$	$2^{-3}=0.125$	$2^{-7}=0.0078125$
$2^4=16$	$2^8=256$	$2^{12}=4096$	$2^{16}=65536$	$2^{-4}=0.0625$	$2^{-8}=0.00390625$

(三)十进制数转换成二进制数

把十进制数转换成二进制数时应将十进制数的整数部分和小数部分分别转换,而且两者的转换方法截然不同。

(1) 整数部分转换方法,把十进制整数依次除 2,记下余数,直到商为 0。最后一次得到的余数是二进制数的最高位,第一次得到的余数是最低位,将所有的余数,从高位到低位依次排列,就是二进制数。这种方法称为“除 2 取余”法。

(2) 小数部分转换方法,把十进制小数依次乘 2 取整数部分,直到小数部分为零。这种方法称为“乘 2 取整”法。

【例 1-3】 $(25.3125)_{10} = (?.?)_2$

解: 对于 $(25)_{10}$, 采用竖式进行计算和转换如下:

		余数	(权)	
$2 \overline{) 25}$	—	1	(2^0)	↓ 低位 高位
$2 \overline{) 12}$	—	0	(2^1)	
$2 \overline{) 6}$	—	0	(2^2)	
$2 \overline{) 3}$	—	1	(2^3)	
$2 \overline{) 1}$	—	1	(2^4)	
0				

直至商=0 为止。

$$(25)_{10} = (11001)_2$$

对于 $(0.3125)_{10}$ ，采用竖式进行计算和转换如下：

			0.3125
		×	2

			0.6250
		×	2

			1.250
		×	2

			0.50
		×	2

			1.0

$$(0.3125)_{10} = (0.0101)_2$$

得出： $(25.3125)_{10} = (11001.0101)_2$

在依次乘 2 取整的过程中，第一次得到的整数部分是小数点后的最高位，将取得的整数从高位向低位顺序排列，就是相应的二进制小数。

应注意：十进制小数依次乘 2 取整，并非都能最终得到小数部分为零。例如把十进制小数 0.42357 转换成二进制数时，“乘 2 取整”可以无限制地进行下去。由于受到计算机精度的限制，只能取有限的位数，这样转换成的二进制数只是一个近似值，若取小数点后 5 位，则得到：

			0.42357
		×	2

			0.84714
		×	2

			1.69728
		×	2

			1.38856
		×	2

			0.77712
		×	2

			1.55424

所以 $(0.42357)_{10} \approx (0.01101)_2$

因为 $(0.01101)_2 = 0 + 0.25 + 0.125 + 0 + 0 + 0.03125 = (0.40625)_{10}$

误差 $\epsilon = 0.42357 - 0.40625 = 0.01732$

由上看出, 小数部分舍去的误差 $\epsilon < 2^{-5}$ 。所以被转换成二进制的位数越多, 其转换的精度越高。

(四) 十进制数转换成十六进制数

同样, 采取的方法也是整数部分和小数部分分别转换。

整数部分转换方法: 除 16 取余;

小数部分转换方法: 乘 16 取整。

【例 1-4】 试将十进制数 $(47632.78125)_{10}$ 转换成十六进制数。

解: $(47632.78125)_{10} = (BA10.C8)_{16}$

十进制数转换成十六进制数, 一般情况下不采用上述方法。而是先将十进制数转换成二进制数, 再由二进制数转换成十六进制数。

三、十进制数的二进制编码及 ASCII 码

(一) 十进制数的二进制编码

人们习惯于用十进制计数, 计算问题的原始数据大多是十进制数。但是采用二进制的计算机不能直接输入或输出十进制数, 为此要用二进制数为十进制数编码, 这就是所谓的二-十进制码或称 BCD 码 (Binary Coded Decimal)。

BCD 码的编码原则是:

(1) 每一位十进制数用 4 位二进制数来表示。如 4 位二进制数的权分别为 8、4、2、1 (称 8421 码制)。这样的 4 位二进制数能编出 16 个码, 但 BCD 码只采用 0~9 十个数码, 其他六个 (A~F) 数码是非法的。

(2) 每位 BCD 码 (4 位二进制数) 之间是逢十进一。

【例 1-5】 试用 BCD 码表示十进制数 896。

解: 十进制数 896 用 BCD 码表示为

$$(896)_{10} = (1000\ 1001\ 0110)_{BCD}$$

注意: BCD 码与二进制数的区别。896 的二进制数为

$$(896)_{10} = (1110000000)_2$$

(二) BCD 码的二-十进制修正

首先要说明计算机对 BCD 码运算仍遵循二进制数的运算规则。当两个 BCD 码相加时, 任一位的“和”可能是 0~9 或者 10~18 之间的某一个值 (考虑到有进位时应为 10~19)。若是 0~9 之间的一个值, 这是正确的; 但若是 10~19 之间的一个值, 这个值对 BCD 码是非法的, 这时对运算的结果必须进行二-十进制修正。现举例说明如下。

【例 1-6】 两个 2 位 BCD 码 $A=31H$, $B=56H$

求 $A+B=?$

解: 按二进制加法规则运算如下: