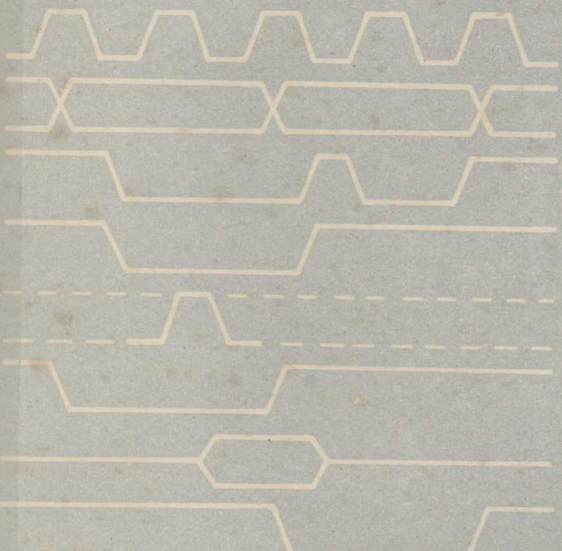
# 微型计算机原理

梁 友 欧阳子炽 编



```
LD    A, 44H
ADD   A, A
LD    B, A
LD    A, 10H
ADD   A, A
ADD   A, A
ADD   A, B
HALT
```

华南理工大学出版社

# 微型计算机原理

梁 友 欧阳子炽 编



华南理工大学出版社

## 内　容　简　介

本书从高师物理专业该课程教学目的和要求出发，以Z 80系列芯片为背景，循序渐进、由浅入深地介绍微型计算机的基本组成、内部信息流通原理、汇编语言程序设计、输入输出及简单的接口应用。其内容比较简明扼要，通俗易懂，便于自学。既可作为师范类物理专业该课程的教材，也可作为物理教师和要求对微型计算机结构和工作原理有一定了解的其他学科师生或其他人员作为自学读本或培训班教材。



**微型计算机原理**

梁　友　欧阳子炽　编

＊

华南理工大学出版社出版发行

（广州　五山）

中国人民解放军海军南海舰队印刷厂（4232工厂）印刷

＊

开本 787×1092　1/16　印张 17.5　字数 536千

1989年1月第1版　　1989年1月第1次印刷

印数　1—3 000

ISBN 7—5623—0117—4/TP·13（课）

定　价：3.50元

# 前　言

微型计算机的应用，可大致分为两大类：一是数据处理，二是实时或定时的检测、控制。虽然对微型计算机硬件结构和内部信息流通原理不甚了解也可以使用计算机高级语言进行数据处理方面的应用。但是如果是维护修理微型计算机和微机化仪器仪表或开发微型计算机于检测控制，由于其与微型计算机内部硬件有较直接的联系，所以需要对微型计算机内部结构、内部信息流通原理、汇编语言和接口技术有较多一点的了解。当然，即使是使用高级语言进行数据处理方面的应用，了解一点计算机原理也是有好处的，可以更好地利用计算机的软件和硬件资源，更得心应手地使用计算机。作为物理教师和很多基层单位的计算机使用者，对微型计算机两方面的应用都可能碰到。此外，电子计算机已成为电子技术应用的一个重要领域，如果不懂得一点微机结构原理方面的知识，在知识结构上也是一个缺陷。所以近几年来，包括师范专科学校在内的不少高等师范院校都开出了介绍微型计算机原理方面的课程。

这本教材就是为了普及微型计算机原理方面知识的需要，根据广东省师专微机·电子技术教研会的决定和分工，并由省高教局组织编写的。编者参考了广东省师专物理专业《微机原理》课程的推荐大纲，在自编油印讲义的基础上，经过五、六轮试用，征求意见，多次修改才最后定稿的。在编写过程中，我们努力按照“打好基础，精选内容，加强实践，体现教法，方便教学”的原则安排全书的结构和进行内容取舍。本书从师专的实际情况出发，以Z80 CPU系列芯片为背景，循序渐进、由浅入深地介绍微型计算机的基本组成，内部信息的流通原理，指令系统，汇编语言程序设计，输入输出及简单的接口技术和应用。为了扩大本教材的适用范围并扩大知识面，还安排了对数字逻辑的介绍及对APPLE Ⅱ微型计算机结构的分析。书中打“*”号的内容可以不讲授，这样，本教材参考教学时数为60学时左右。

本教材由雷州师专梁友和惠阳师专欧阳子炽合作编写，梁友为主编。欧阳子炽编写了第六至第九章和附录三的初稿并参加了对书稿的讨论和审定；梁友编写了绪言、第一至第五章和附录一至二以及第六至第八章部分内容，全书由梁友统稿和定稿。

我们在编写这本书时参考了很多教材和专著，并引用了其中一些文献的资料、图表和观点。此外，在编写过程中，中山大学电子学系林贻坤教授、华南师大龙启钧副教授、佛山大学教务处周奉年副教授对书稿提出了很多宝贵的意见。本书的出版得到了广东省高教局教学处、物资站以及雷州师专教务处、学报室、物理系等部门很多同志的大力支持，在此一并致以衷心感谢。

虽然我们在编写中做了很大的努力，但由于编者水平和经验等原因，本书肯定有不少错漏和不足，恳请同行专家和广大读者给予指正。

编　　者

一九八八年八月于湛江

# 目　录

# 绪　言

电子计算机是本世纪人类最辉煌的科学技术成就之一。随着信息处理技术的发展，电子计算机正从深度和广度两个方面渗透到人类社会的各个领域，它的应用范围已从最初的科学计算发展为对各种信息的加工处理。计算机的普及应用，是现代化社会的重要标志之一。

世界上第一台电子计算机"埃尼阿克"（ENIAC）是一九四六年制成的。它用了18,800个电子管，1,500多个继电器，全机重三十吨，体积85立方米，占地150平方米，耗电150千瓦，每秒运算5000次，造价100多万美元。此后三十多年，计算机科学的发展十分迅速。按照组成计算机的电子器件来划分，至今已经历了电子管、晶体管、中小规模集成电路和大规模集成电路四代的演变，差不多十年左右就要换一代（见表0－1）。近年来，一些国家正在加紧研制第五代计算机。可以预言，随着半导体技术、激光和光纤技术、超导技术的发展，综合所有新技术的性能更加完善的电子计算机，迟早会出现在我们的面前。

通用电子计算机

表0－1

| 代　数 | 时　期 | 器　件 | 代表产品 |
|---|---|---|---|
| 第一代 | 1945～1959 | 电子管 | IBM650 |
| 第二代 | 1960～1964 | 晶体管 | IBM7090 |
| 第三代 | 1965～1969 | 中小规模集成电路 | IBM360/370 |
| 第四代 | 1970～ | 大规模集成电路 | 303X、4300系列 |
| 第五代 | 1980～ | 超大规模集成电路 | 在研制中 |

电子计算机有模拟机和数字机两类。简单地说，前者处理连续变化的量，而后者则处理离散的不连续的量。目前世界上数字计算机的数量远远超过模拟计算机。通常，我们所讲的计算机是指数字计算机。

电子数字计算机的核心部分是由一些能够表示两种状态（例如用3V左右的高电位表示"1"状态，用0.3V左右的低电位表示"0"状态）、具有一定逻辑功能的逻辑电路（如各种门电路、触发器、寄存器、译码器……等等）所组成。因而这种计算机要处理的所有信息，包括要求它工作的命令，各种数值、符号和字母，甚至汉字等，都要用"1"和"0"的组合代码来表示。每组"1"和"0"的组合代码均可看成是一个二进制数。这就意味着电子计算机的输入信息和输出信息均是一些二进制数（其本质是一连串的高低电平）。电子计算机就是能高速度地，按照人们事先规定的步骤自动地对数字进行加工和处理的机器。由于电子计算机具有记忆（或贮存）的信息量大（例如一个只有指甲大小的集成电路芯片可以存放几十万、几百万甚至更多的"0"和"1"，一个只有唱片大小的磁盘可以存放成百兆个"0"和"1"，而这些"0"和"1"的组合将可以表示相当大的信息量），运算速度高（目前已有每秒运算十亿次以上的计算机），运算准确，出错机会少，以及可以作出各种逻辑判断（例如可以判断数的正负、大小以及信息的有无）等特点，使得它能极大地扩展人脑的功能，可以在较短的时间内完成一个人一辈子也完成不了的工作量。例如，十八世纪英

国数学家商克斯用“手算”，花了二十年时间才算到园周率π的707位(其中527位还有误)，而现在一台高速度的计算机只需28小时就可算到π的2936万位。

电子计算机发展的其中一个明显趋势是要求体积小，价格低。以大规模集成电路为基础的微型计算机由于具有体积小，性能价格比最优，耗电小，特别是价格低廉这些特点，因而自1971年问世以来，就得到极其迅速的发展。其发展之迅速，影响之深远，远远超过了它的前代。现在一小片微型机的功能，超过了五十年代初期占地上百平方米，耗电成百千瓦的电子管计算机；今天十多美元的单片微型机，其性能已达到二十多年前成十万美元的晶体管计算机系统。作为微型计算机核心部件的微处理器，自一九七一年以来，也经历了四代更新，平均每几年左右换代一次（见表0—2）。近十多年来，全世界微型计算机平均年产量递增40％左右（见表0—3），性能价格比急剧上升（据统计，计算机芯片的价格以平均每年递减30％的速度下降）。由微型机阵列构成巨型机的方案已提出了不少。展望九十年代，微型计算机的发展方兴未艾。据预测，几年以后，不仅会出现各种各样的新型计算机，而且在技术上也必定实现硬件和软件产品的标准化、系列化、外部设备的多样化以及以微型计算机为主体的网络系统和多机系统。微型计算机的出现是计算机发展史上的一个里程碑。随着集成电路工艺的发展，近几年来，人们在一块芯片上集成了包括微处理器、存贮器和多种输入输出接口等功能部件，构成了单片微型计算机（简称单片机）。就其功能来说，这一小片单片机就相当于一台计算机。由于单片机性能价格比更优越，体积、重量大为减少，发展非常迅速，应用十分广泛。其潜在能力已愈来愈为人们所注意。

微 处 理 器

表0—2

| 发表年 | 代 数 | 代表型号 | 集成度 | 位 数 |
|---|---|---|---|---|
| 1971 | 第一代 | 英特尔4004 | 2,200 | 四位机 |
| 1973 | 第二代 | I 8080、M6800、Z 80 | 4,800 | 八位机 |
| 1978 | 第三代 | M68000、NS 1600 | 29,000 | 十六位机 |
| 1981 | 第四代 | NS32032 | 100,000 | 三十二位机 |

**全世界微处理器与微型计算机产量统计**

（单位：10万台） 表0—3

| 年 份 | 77年 | 78年 | 79年 | 80年 | 81年 | 82年 | 83年 | 84年 | 85年 |
|---|---|---|---|---|---|---|---|---|---|
| 微处理器 | 100左右 | 近300 | 568 | 1028 | 1475 | 1964 | 2390 | 2660 | 3100 |
| 微型计算机 | 0.6 | 1.8 | 2.6 | 5.4 | 10 | 20 | 28 | 37 | 46 |

尽管数字计算机“识别”和处理的对象只能是一些二进制数，但由于各种各样的信息都可编码转换成二进制数的形式，这就大大扩充了它的应用范围。概括地说，微型计算机应用可分为两大类：一是数据处理，包括各种科学计算、事务管理、辅助设计、情报检索等；二是实时的检测控制。

一台实用的计算机应包括硬件和软件两大部分。简言之，所谓硬件，是指那些看得见的物质实体；而要求计算机工作的各种程序则属于软件。为了方便用户使用，计算机生产厂家

和计算机科学工作者已研制出了大量软件（系统软件），供用户调用。这些软件和硬件的巧妙配合，使得一般使用者无需知道计算机硬件结构及内部信息流通原理，就可应用计算机高级语言编写用于数据处理方面的应用程序，并根据使用说明书上机操作。但是若应用计算机于实时检测控制，由于其与服务对象有较多的硬件联系，且控制功能上有实时要求，这就要求应用者不但会根据应用对象的特点设计相应的应用程序，还需要了解微型计算机的硬件结构特点和有关的接口技术。对师范院校物理专业学生来说，无论是管理维护和使用微型计算机或智能化仪器仪表，还是开发应用微机（包括单板机或单片机）于实时检测控制、改革教学或实验手段、改制仪器设备，或者在中学开展微机方面的研究和组织中学生开展课外科技活动，都需要对微机的内部结构原理有较多的了解。当然，即使是使用计算机高级语言进行数据处理方面的应用，对微型计算机的工作原理、硬件结构和指令系统有一定的了解，也是有好处的。例如，可以更好地利用计算机的软件和硬件资源，而且通过学习计算机的机器语言指令及利用这种指令编程，能够更加深对高级语言本质的理解，打破对计算机的神秘感。

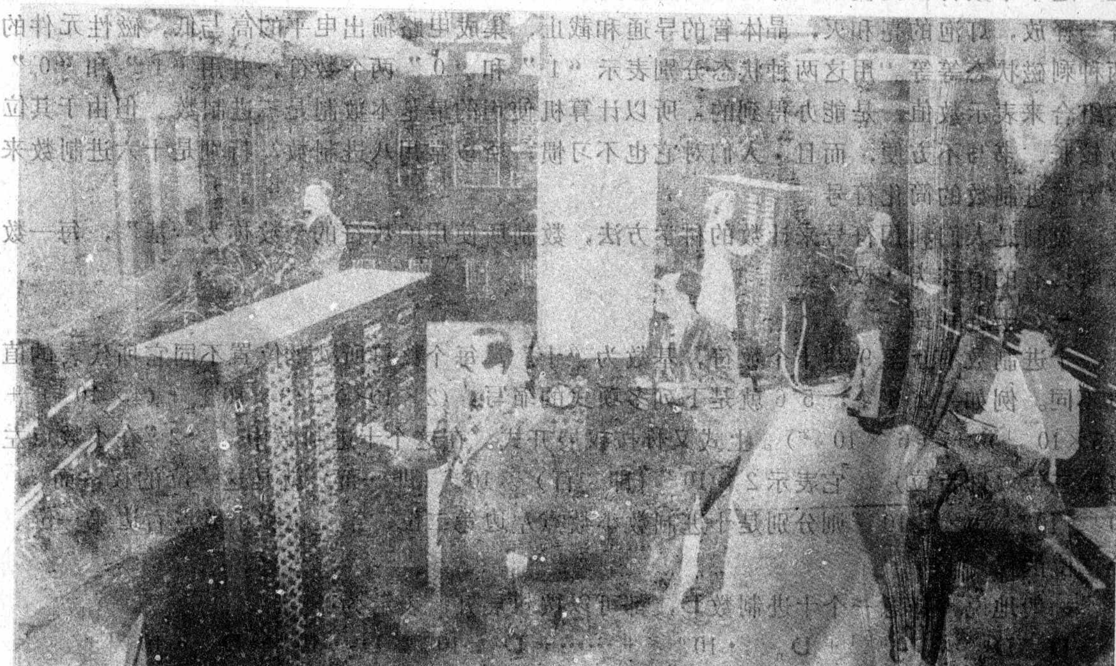


图 0—1　ENIAC——世界上第一台电子数字计算机

# 第一章　电子计算机中的数制和码制

**内容提要：** 本章作为预备知识，介绍计算机常用的二进制和八进制、十六进制、十进制数的表示方法，以及这几种数制之间的转换。还介绍了原码、反码、补码、BCD码和ASCII码的基本概念。

## §1—1　进位计数制

计算机之所以能实现，最基本的一条就是能用一些物理器件的状态来表示数。日常生活中用得最多的是十进制数，它有十个数字的符号：0、1、2、3、4、5、6、7、8、9。十进制数具有很多优点，但是我们很难找到一种具有十种稳定状态的物理器件来表示0至9这十个数符。而使某器件处于两种稳定状态，则是一件容易做到的事。例如继电器的吸合与释放，灯泡的亮和灭，晶体管的导通和截止，集成电路输出电平的高与低，磁性元件的两种剩磁状态等等。用这两种状态分别表示“1”和“0”两个数符，并用“1”和“0”的组合来表示数值，是能办得到的。所以计算机使用的最基本数制是二进制数。但由于其位数较长，书写不方便，而且，人们对它也不习惯，所以常用八进制数，特别是十六进制数来作为二进制数的简化符号。

数制是人们利用符号来计数的科学方法，数制所使用的数符的个数称为“基”，每一数位所具有的值称为“权”。

**一、十进制数**

十进制数有0至9共十个数符，基数为“十”。每个数符所处的位置不同它所代表的值也不同。例如，234.56就是下列多项式的缩写：$(2\times10^{2})+(3\times10^{1})+(4\times10^{0})+(5\times10^{-1})+(6\times10^{-2})$。上式又称按权展开式。在这个十进制数中，“2”在小数点左边第三位（即百位），它表示$2\times10^{2}$（即二百）。$10^{2}$（即一百）就是这一位的权，而$10^{1}$、$10^{0}$、$10^{-1}$、$10^{-2}$则分别是十进制数小数点左边第二位、第一位和小数点右边第一位、第二位的权。

一般地说，任意一个十进制数D，都可以按权展开，表示为：

$$D = D_{n-1}\cdot10^{n-1}+D_{n-2}\cdot10^{n-2}+\cdots\cdots+D_{1}\cdot10^{1}+D_{0}\cdot10^{0}+D_{-1}\cdot10^{-1}+D_{-2}\cdot10^{-2}+\cdots\cdots+D_{-m}\cdot10^{-m}$$

$$=\sum_{i=n-1}^{-m} D_{i}\cdot10^{i} \qquad \cdots\cdots\cdots\cdots (1-1)$$

其中i表示数的某一位；$D_i$表示第i位的数符，它可以是0—9中的任一个，由具体的数D确定；n为小数点左边的位数，m为小数点右边的位数。式中的“10”即十进制的基数。这种按权展开的表示方法，可推广应用于任意进制数，这时只需注意到在不同数制中基数不同以及$D_i$的取值范围不同即可。

**二、二进制数**

二进制数中，只有0和1两个数符，基数为2，进位时“逢二进一”。任一个二进制数B都可按权展开为：

$$B = \sum_{i=n-1}^{-m} B_i \cdot 2^i \quad \cdots\cdots\cdots\cdots (1-2)$$

其中$B_i$只能取 1或 0，由具体的B确定；n 为小数点左边的位数，m为小数点右边的位数。

例如，二进制数1 1 0 1 · 1 1按权展开为：

$$\begin{aligned}(1101\cdot11)_2 &= 1\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}\\ &= 8+4+1+\tfrac{1}{2}+\tfrac{1}{4}\\ &= (13.75)_{10}\end{aligned}$$

【注：上式括号外的下标“2”和“10”分别表示括号内的数是二进制和十进制数——下面也用类似的办法表示其他进位制数】

由此可见，从小数点算起，左边各位的权分别为1，2，4，8，…… 右边各位的权分别为$\frac{1}{2}$，$\frac{1}{4}$……

二进制数的四则运算很简单：

1.加法时需注意逢二进一，其余同十进制。例如：

①0 + 0 = 0

②0 + 1 = 1 + 0 = 1

③1 + 1 = 10（逢二进一）

2.减法时需注意借一当二，其余同十进制。例如：

①0 − 0 = 0

②1 − 0 = 1

③1 − 1 = 0

④10 − 1 = 1（借一当二）

3.乘法时，其法则同十进制，但在部分积相加时，要注意逢二进一。例如：

①0 × 0 = 0

②0 × 1 = 1 × 0 = 0

③1 × 1 = 1

④11 × 11 = 1001

4.除法运算方法基本同十进制，但实际运算时应注意借一当二。例如：10110100 ÷ 1001的计算方法如下：

```
              1 0 1 0 0
        ┌─────────────────
1 0 0 1 │ 1 0 1 1 0 1 0 0
          1 0 0 1
          ───────
            1 0 0 1
            1 0 0 1
          ───────────────
                  0 0 0
```

由此计算出10110100 ÷ 1001 = 10100。

## 三、八进制数

八进制有0、1、2、3、4、5、6、7，共八个数符，基数为8，进位时“逢八进一”。任一个八进制数Q，都可以按权展开表示为：

$$Q = \sum_{i=n-1}^{-m} Q_i \cdot 8^i \qquad \cdots\cdots\cdots\cdots (1-3)$$

其中Qi 可取0 — 7 之间的值，取决于数值Q；n为小数点左边的位数，m为小数点右边的位数。

八进制数从小数点算起，左边各位的权分别为1，8，64…… 右边各位的权分别为 $\frac{1}{8}$，$\frac{1}{64}$，$\frac{1}{512}$……

八进制数进行四则运算时，应注意逢八进一和借一当八。

## 四、十六进制数

十六进制数的基数是16，它有十六个数符，除了0至9这十个数字外，还有六个数符用字母A、B、C、D、E、F表示，它们分别代表十进制数的10、11、12、13、14、15。十六进制数进位时是“逢十六进一”；减法借位时是“借一当十六”，任一个十六进制数H，均可按权展开为：

$$H = \sum_{i=n-1}^{-m} H_i \cdot 16^i \qquad \cdots\cdots\cdots\cdots (1-4)$$

其中$H_i$ 可取0 —— F之间的值，取决于数值H(但要注意，在将十六进制数转换成十进制数时，要用十进制数值10——15分别代替 A—— F这六个字母代入计算)；n为小数点左边的位数，m为小数点右边的位数。

十六进制、十进制、八进制、二进制数的对照如表1—1所列。

各种数制的对照表

表1—1

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

## 五 、几种进位制数表示法

前面已介绍了十进制、二进制、八进制、十六进制等表示数值的方法。因此书写时如表示得不明确，会造成混乱。除了如前所述用下标的办法外，常在数字前面或后面加符号来区分，如表1—2。

**表示进位制的符号** 表1—2

| 进位制 | 写在数字前面 | | 写在数字后面 | |
|---|---|---|---|---|
| | 符号 | 例 | 符号 | 例 |
| 十进制 | 无符号 | 98 | D 或 无符号 | 98或98D |
| 十六进制 | $ | $7F | H | 7FH |
| 八进制 | @ | @275 | O 或 Q | 275Q |
| 二进制 | % | %11010 | B | 11010B |

## §1—2 进位制数之间的转换

### 一、二进制、八进制、十六进制数转换成十进制数

二进制、八进制、十六进制数转换成十进制数的方法很简单。根据（1—2）、（1—3）、（1—4）式，即根据按权展开的方法，很方便地将二、八、十六进制数转换成十进制数。例如：

$(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (13)_{10}$

$(27)_8 = 2 \times 8^1 + 7 \times 8^0 = (23)_{10}$

$(3A)_{16} = 3 \times 16^1 + A \times 16^0 = 3 \times 16^1 + 10 \times 16^0 = (58)_{10}$

### 二、十进制数转换成二进制数

十进制数转换成二进制数，一般要把整数部分和小数部分分开来处理，然后再将两部分用小数点连起来。

1.十进制整数转换成二进制数

其方法：先把此十进制整数连续被基数2来除，依次记下余数，直到不能整除为止。第一个余数即为整数部分的最高位。这种方法简称为“除（以）2取余”。下面以一例子来说明。

| 二进制基数 | 十进制数 | 余数 | |
|---|---|---|---|
| 2 | 12 | ………0 | ←二进制数整数部分的最低位 |
| 2 | 6 | ………0 | |
| 2 | 3 | ………1 | |
| | 1 | ………1 | ←二进制数整数部分的最高位 |

于是十进制数12被转换为二进制数1100，即

$(12)_{10} = (1100)_2$

2.十进制小数部分转换成二进制数

其方法是：把十进制数的小数部分乘以基数2，乘积的整数部分（0或1）即为所求二

进制小数的最高位。然后再把乘积的小数部分乘以2（注意：乘积的整数部分不要再乘以2），这一次乘积的整数部分（0或1）即为二进制小数部分次高位。依此类推直到小数部分已为0或达到所需的精度为止（考虑到并不是所有十进制小数都能用有限位二进制小数精确表示，这就需要根据一定的精度要求进行取舍）。这种方法简称为"乘2取整"。下面以一例子来说明。

乘积整数部分 ↓　　十进制数 ↓

0.6875
× 2 ← 基数

二进制数小数部分的最高位 → 1 .3750
× 2

0 .7500
× 2

1 .5000
× 2

二进制小数部分的最低位 → 1 .0000

这样，便将十进制小数0.6875转换成了二进制小数0.1011，即 $(0.6875)_{10}=(0.1011)_2$。从上面的分析，不难看出十进制数12.6875等于二进制数1100·1011，即

$(12.6875)_{10}=(1100.1011)_2$

至于十进制数转换成八进制数或十六进制数，其方法与十进制数转换成二进制数的方法类同，不过需注意八进制和十六进制数的基数分别为8和16。读者可自行分析之。

**三、二进制数与八进制数的相互转换**

1.二进制数转换成八进制数

三位二进制数恰好有八种组合（000,001,……111），因此，每一个八进制数符，可以用三位二进制数表示。在二进制数转换成八进制时，可以从小数点开始，向左和右分别把整数和小数部分每三位分成一组（整数部分最左边一组是否足三位，也可不管它，但小数部分最右一组不足三位时，则应在其右边加0补足到三位），然后用一等值的八进制数符代换每一组二进制数。现举例说明如下：

设有一个二进制数10110101·01001101，从小数点开始向左和向右把每三位分为一组：

10　110　101·010　011　01（0）←右边加0补足三位

再把每一组二进制数转换成八进制数：

10　110　101·010　011　01（0）
↑　↑　↑　↑　↑　↑
2　6　5·2　3　2

这样，便把二进制数10110101·01001101转换成了八进制数265·232，即

$(10110101\cdot01001101)_2=(265\cdot232)_8$

2.八进制数转换成二进制数

其方法是从高位到低位依次用三位二进制数等值地代替每一位八进制数，并把它们连起来（连起来后，二进制数部分最左边的0和小数部分最右边的0一般也可以省写）　例

如，可用此法将八进制数307·164转换成二进制数：

$$\begin{array}{ccccccc} 3 & 0 & 7 & \cdot & 1 & 6 & 4 \\ \uparrow & \uparrow & \uparrow & & \uparrow & \uparrow & \uparrow \\ 011 & 000 & 111 & \cdot & 001 & 110 & 100 \end{array}$$

可见，$(307\cdot164)_8=(11000111\cdot0011101)_2$

**四、二进制数与十六进制数相互转换**

1.二进制数转换成十六进制数

四位二进制数能得到16种组合，因此，可用类似于二进制数转换成八进制数的方法，将二进制数转换成十六进制数：从小数点开始，向左和向右分别把二进制数的整数和小数部分每四位分成一小组（整数部分最左边一组是否足四位，也可不管它，但小数部分最右一组不足四位时，则必须在其右边加 0补足到四位），然后用等值的十六进制数符代换每一组二进制数（等值的方法可参照表 1—1）。例如，可用此法将二进制数1001010·011101 转换成十六进制数：

$$(1001010\cdot011101)_2=(\underline{100}\,\underline{1010}\cdot\underline{0111}\,\underline{0100})_2$$

$$\begin{array}{ccccc} \uparrow & \uparrow & & \uparrow & \uparrow \\ 4 & A & \cdot & 7 & 4 \end{array}$$

$$=(4A\cdot74)_{16}$$

2.十六进制数转换成二进制数

其方法亦类似八进制数转换成二进制数，但要注意到每个十六进制数符要用一组等值的四位二进制数代换。下面是一个进行这种转换的例子：

$$(25\cdot C8)_{16}=(\underline{0010}\,\underline{0101}\cdot\underline{1100}\,\underline{1000})_2$$

$$\begin{array}{ccccc} \uparrow & \uparrow & & \uparrow & \uparrow \\ 2 & 5 & \cdot & C & 8 \end{array}$$

$$=(100101\cdot11001)_2$$

## §1—3　二进制编码

计算机只“认识”二进制数，因此，要在计算机中表示的数字、字母、符号甚至汉字等等都要以特定的二进制代码来表示，这就是二进制编码。本节分别介绍十进制数符的二进制编码和字母与符号的二进制编码。

**一、二进制编码的十进制数**

这种编码的指导思想是用四位二进制数来表示一位十进制数（即用四位二进制数来表示一个十进制数符）。

由于四位二进制数有十六种不同组合，而我们只需其中的十种组合表示十个十进制数符。可见，编码的方法可以有很多种。不过，人们使用得最多的是“8421BCD码”，在这种编码方式中，每四位二进制数与十进制数字的对应关系如表1—3。

BCD编码表　　　　　　　　表1—3

| 十进制 | 二进制 | 说　　明 | 十进制 | 二进制 | 说　明 |
|---|---|---|---|---|---|
| 0 | 0000 | | 8 | 1000 | |
| 1 | 0001 | | 9 | 1001 | |
| 2 | 0010 | | | 1010 | 非法码（不能用） |
| 3 | 0011 | | | 1011 | |
| 4 | 0100 | | | 1100 | |
| 5 | 0101 | | | 1101 | |
| 6 | 0110 | | | 1110 | |
| 7 | 0111 | | | 1111 | |

表1—3中的二进制数，每一位都有特定的“权”。从左到右各位的权依次为$2^3=8$，$2^2=4$，$2^1=2$，$2^0=1$，所以这种编码称为8421BCD码，简称BCD码。

BCD码通常用于微机仪表化的输入输出。

将BCD码转换成十进制数时要注意到每四位为一组，对应一个十进制数符。而将十进制数转换成BCD码时，要注意十进制的每一个数符对应一组四位二进制数。千万不要把十进制数等值的二进制数与这个十进制数的BCD码两个概念搞混了。请看下面一个例子：

根据十进制数转换成等值的二进制数的方法，可把十进制数57转换成二进制数。

$$(57)_{10}=(111001)_2$$

但十进制数57的BCD码却是：

$$(57)_{10}=(\underbrace{0101}_{5}\ \underbrace{0111}_{7})_{BCD}=(01010111)_{BCD}$$

显然，二者是不同的。

## 二、字母与字符的编码

如前所述，字母和各种字符都必须按特定的规则用二进制编码才能在计算机中表示。编码的方式很多，但目前在微机中最普遍使用的是ASCII码（American Standard Code for Information Interchange美国标准信息交换码）。ASCII码把包括英文26个大写字母，26个小写字母，0～9共十个数字，还有一些专用字符如！、％等以及控制符号如换行、换码、回车等等共一百二十八个符号，用7位二进制数来编码，称为全ASCII码。其编码表见表1—4。

**ASCII码表**　　　　　　　　　　　　　　　　　　　　　　　　表 1—4

| 字符 位 3210 \ 位 654 | | 0<br>000 | 1<br>001 | 2<br>010 | 3<br>011 | 4<br>100 | 5<br>101 | 6<br>110 | 7<br>111 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0000 | NUL | DLE | SP | 0 | @ | P | 、 | p |
| 1 | 0001 | SOH | DC1 | ! | 1 | A | Q | a | q |
| 2 | 0010 | STX | DC2 | ” | 2 | B | R | b | r |
| 3 | 0011 | ETX | DC3 | # | 3 | C | S | c | s |
| 4 | 0100 | EOT | DC4 | $ | 4 | D | T | d | t |
| 5 | 0101 | ENQ | NAK | % | 5 | E | U | e | u |
| 6 | 0110 | ACK | SYN | & | 6 | F | V | f | v |
| 7 | 0111 | BEL | ETB | ’ | 7 | G | W | g | w |
| 8 | 1000 | BS | CAN | ( | 8 | H | X | h | x |
| 9 | 1001 | HT | EM | ) | 9 | I | Y | i | y |
| A | 1010 | LF | SUB | * | : | J | Z | j | z |
| B | 1011 | VT | ESC | + | ; | K | [ | k | {{ |
| C | 1100 | FF | FS | , | < | L | \ | l | \| |
| D | 1101 | CR | GS | − | = | M | ] | m | }} |
| E | 1110 | SO | RS | . | > | N | Ω | n | ~ |
| F | 1111 | SI | US | / | ? | O | — | o | DEL |

其中：

NUL　空
SOH　标题开始
STX　正文开始
ETX　正文结束
FOT　传输结束
ENQ　询问
ACK　承认
BEL　报警
BS　退格
HT　横向列表
LF　换行
VT　纵向列表

FF　走纸控制
CR　回车
SO　换出字符（移出）
SI　换入字符（移入）
DLE　数据链换码
DC1　设备控制1
DC2　设备控制2
DC3　设备控制3
DC4　设备控制4
NAK　否定
SYN　同步空转
ETB　信息块传送结束

CAN　作废
EM　纸尽
SUB　代替字符
ESC　换码
FS　文件分隔符
GS　组分隔符
RS　记录分隔符
VS　单元分隔符
SP　空格
DEL　删除

此外还有一种ASCII码，它去掉了26个小写英文字母。

如果用8位二进制数其中7位表示ASCII码，多余的最高一位常用于检查错误，称为奇