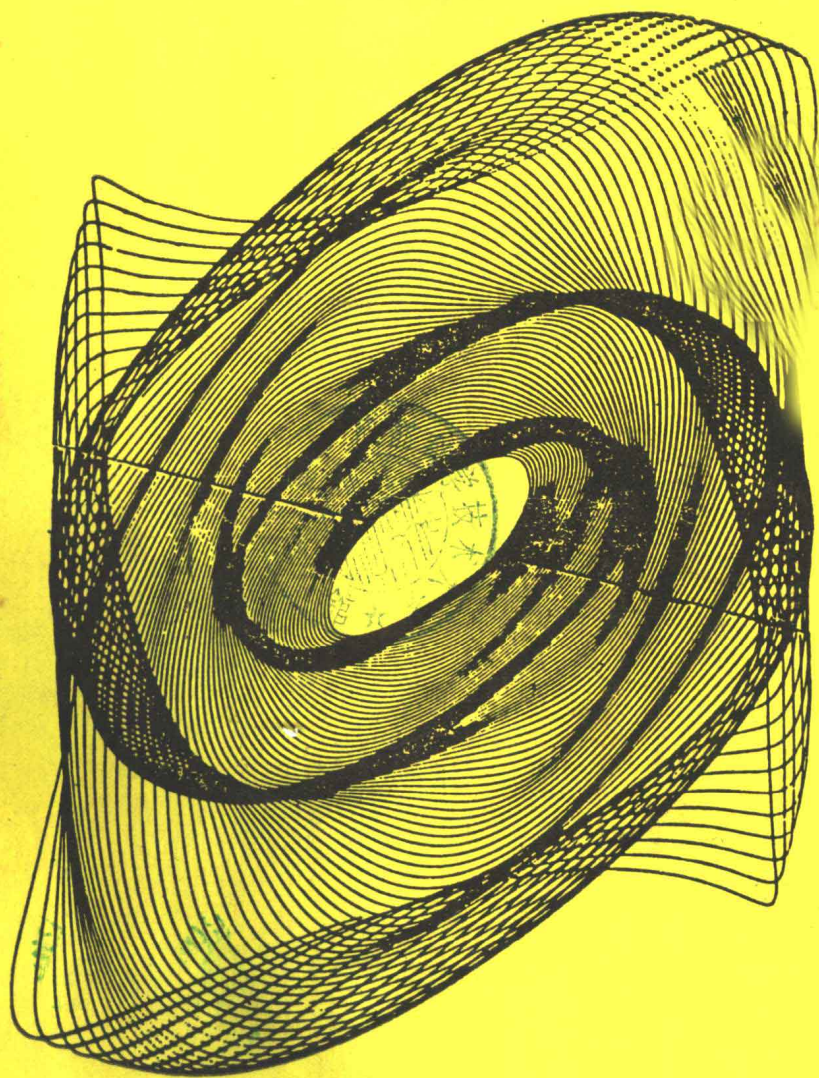


扩充BASIC语言与 程序剖析

程光钺 葛履明 编著



四川教育出版社



扩充 BASIC 语言与程序剖析

程光铎
葛履明 编著

四川教育出版社

一九八六年·成都

扩充 BASIC 语言与程序剖析

四川教育出版社出版	(成都盐道街三号)
四川省新华书店发行	温江县印刷厂印刷
开本787×1092 1/16	印张18字数395千
1987年2月第一版	1987年2月第一次印刷
印数: 1—1,500册	
书号: 7344·586	定价3.97元

前 言

本书作者在《BASIC语言基础及PC-1500机》一书(四川教育出版社, 1985年出版)中介绍了BASIC语言的最基本部分, 它大致相当于国家标准GB4144-84中的最小BASIC (Minimal BASIC)。本书介绍扩充BASIC语言, 并着重讨论程序的剖析方法。

本书有以下特点:

1. 采用严谨的语法体系

与惯常讲述BASIC语言的方法不同, 本书采用了一套严谨的语法(包括词法与句法)体系, 能使读者更系统和深入地掌握这种程序设计语言。

2. 突出程序剖析

学习计算机语言和程序设计, 不仅要学习一定的基本知识, 更重要的是学习运用这些基本知识的方法。程序剖析就是其中最重要的方法之一。学习如何剖析程序, 是为了看懂别人的程序, 掌握其来龙去脉, 借鉴其思路与技巧, 从而得心应手地编写自己的程序。本书将程序剖析的讨论贯穿全书。

3. 讨论程序与数据在机内的存放及其应用

一般介绍BASIC语言的书籍并不涉及程序与数据在计算机内的存放方式。由于对系统内部情况不清楚, 编程具有一定的盲目性, 无法预先估计所需内存占用量及其分配, 无法在更高层次上节省内存与提高程序的质量。另一方面, 有关计算机软件和结构的书籍, 虽然对于上述问题有深入的讨论, 但往往要求准备知识较多, 一般读者难以直接与其编程实践相联系。有鉴于此, 本书以一定的篇幅, 针对PC-1500机并结合编程介绍这方面的问题。讨论立足于应用, 并采用一般读者能接受的方式加以阐述。

4. 发掘PC-1500机的绘图与显示功能

PC-1500便携式微型机是国内目前使用台数最多的微机, 但其功能还未被充分发掘。绘图功能是该机的重要特色, 本书对此详加讨论, 并配以大量的应用实例。屏幕显示虽非该机特色, 但因这方面的资料较少, 本书亦进行较详细的介绍。相应的绘图、显示语句虽然对于不同计算机(绘图仪)不尽相同, 但本书所介绍的方法具有普遍意义。

5. 起点低、终点高

本书起点低, 文字通俗, 解释详细, 深入浅出, 用于指导自学基本上可做到无师自

通。本书从计算机应用来说终点高，对有一定基础的读者是扩大知识、深入学习的好参考书。在实例选择方面注意了趣味性，以提高读者阅读的兴趣。

本书的初稿几年来曾多次在四川大学、成都科技大学、中国科学院成都分院化学所、地理所等单位，作为讲授计算机应用课程的教材或指导自学使用。

本书针对PC-1500机，对其BASIC语言（简称BASIC-1500）、显示、绘图、文件存取的各种情况分门别类地加以讨论。因此对该机的用户，此书亦可用作手册。书前有详细的目录，书后（附录一）将各语法词汇按字母排列，并给出与之相应的章节，查询检索十分方便。

本书由四川大学计算机系张升楷副教授审阅，在此表示衷心感谢、
由于作者水平有限，书中如有不妥之处，敬请批评指正。

目 录

前 言	(1)
第一章 扩充BASIC语言基本词法	(1)
1.1 基本符号	(1)
1.2 语法词汇	(1)
1.3 常量	(2)
1. 数常量	(2)
(1) 十进制	(2)
(2) 二进制	(3)
(3) 十六进制	(4)
(4) r进制	(6)
(5) 二——十进制 (BCD码)	(10)
(6) 二进制的原码和补码表示	(10)
(7) 数在计算机内的存放方式	(13)
2. 字符串常量	(14)
(1) 一般介绍	(14)
(2) 字符串在计算机内的存放方式	(14)
3. 逻辑值常量	(15)
1.4 变量	(15)
1. 数变量	(16)
(1) 简单数变量	(16)
(2) 下标数变量	(17)
(3) 数变量在计算机内的存放方式	(18)
2. 字符串变量	(20)
(1) 简单字符串变量	(20)
(2) 下标字符串变量	(20)
(3) 字符串变量在计算机内的存放方式	(21)

3. 逻辑变量	(23)
1.5 函数	(23)
1. 数函数	(25)
(1) 代数函数	(25)
(2) 三角函数与反三角函数	(30)
(3) 角度(时间)单位换算函数	(31)
(4) 时间函数(TIME)	(31)
(5) 随机函数(RND)	(34)
2. 字符串函数	(36)
(1) 字长函数(LEN)	(36)
(2) 字码转换函数(ASC)	(37)
(3) 字数转换函数(VAL)	(37)
(4) 码字转换函数(CHR\$)	(38)
(5) 数字转换函数(STR\$)	(38)
(6) 求左部函数(LEFT\$)	(38)
(7) 求右部函数(RIGHT\$)	(39)
(8) 求中部函数(MID\$)	(39)
(9) 字符键入函数(INKEY\$)	(41)
3. 内存函数与计算机的内存分配	(42)
(1) 计算机内存分配	(42)
(2) BASIC程序在计算机内的存放	(44)
(3) 内存函数STATUS1, STATUS2 及STATUS0(MEM)	(45)
(4) 数据在计算机内的存放	(46)
(5) 内存函数STATUS3	(47)
(6) PEEK函数	(49)
(7) POKE命令	(50)
1.6 表达式	(52)
1. 第一种算术表达式	(53)
(1) 算术初等量	(53)
(2) 算术运算符	(53)
(3) 第一种算术表达式	(53)
(4) 运算的优先次序	(54)
(5) 数关系运算符	(54)
(6) 第一类关系式	(55)
(7) 小结	(55)
2. 字符串表达式	(56)

(1) 字符串初等量	(56)
(2) 字符串运算符	(56)
(3) 字符串表达式	(56)
(4) 字符串关系运算符	(56)
(5) 第二类关系式	(57)
(6) 小结	(57)
3. 逻辑表达式	(58)
(1) 逻辑初等量	(58)
(2) 逻辑运算符	(58)
(3) 逻辑表达式	(58)
(4) 小结	(59)
4. 第二种算术表达式	(59)
(1) AND 运算	(59)
(2) OR 运算	(60)
(3) NOT 运算	(61)
(4) 小结	(62)
第二章 扩充BASIC语言基本句法	(65)
2.1 一般介绍	(65)
2.2 赋值与输入类语句	(67)
1. 赋值 (LET) 语句	(67)
2. 键盘输入 (INPUT) 语句	(69)
3. 读/置数据 (READ/DATA) 语句	(72)
4. 恢复数据 (RESTORE) 语句	(73)
5. 自动读数据 (AREAD) 语句	(74)
6. 整机清零 (CLEAR) 语句	(75)
2.3 显示输出类语句	(76)
1. 显示 (PRINT) 语句	(76)
2. 暂显 (PAUSE) 语句	(79)
3. 等待显示时间选择 (WAIT) 语句	(80)
4. 清屏 (CLS) 语句	(83)
5. 其它语句	(85)
2.4 打印与发声类语句	(85)
1. 字体控制 (CSIZE) 语句	(85)
2. 打印 (LPRINT) 语句	(86)
3. 格式 (USING) 语句	(90)
4. 走纸 (LF) 语句	(92)

5. 颜色选择 (COLOR) 语句	(93)
6. 发声 (BEEP) 语句	(93)
7. 声开 (BEEP ON) 与声闭 (BEEP OFF) 语句	(95)
2.5 转向与条件语句	(95)
1. 转向 (GOTO) 语句	(95)
2. 条件 (IF) 语句	(99)
3. 开关转向 (ON GOTO) 语句	(103)
4. 遇错转向 (ON ERROR GOTO) 语句	(106)
2.6 循环	(108)
1. FOR语句与NEXT语句	(109)
2. 循环体语句	(111)
3. 实例	(115)
2.7 子程序	(122)
1. 子程序段与主程序段	(122)
2. 转子 (GOSUB) 语句	(124)
3. 返回 (RETURN) 语句	(124)
4. 开关转子 (ON GOSUB) 语句	(127)
5. 实例	(128)
2.8 其它语句与命令	(137)
1. 程序的解释 (REM)	(137)
2. 程序的暂停 (STOP) 与继续 (CONT)	(137)
3. 状态冻结 (LOCK) 和解冻 (UNLOCK)	(138)
4. 程序的跟踪 (TRON) 和去跟踪 (TROFF)	(138)
第三章 文字与图形的显示	(145)
3.1 显示屏	(145)
1. 显示屏的显示方法	(145)
2. 显示屏横向按位标识法	(145)
3. 显示屏横向按列标识法	(145)
4. 显示屏纵向十进制标识法	(146)
5. 显示屏纵向十六进制标识法	(146)
3.2 与显示有关的语句及函数	(147)
1. 按位移光标 (CURSOR) 语句	(147)
2. 按列移光标 (GCURSOR) 语句	(148)
3. 按列显示 (GPRINT) 语句	(148)
4. 显示数值化 (POINT) 函数	(152)
3.3 自选内容的静态显示	(153)

1. 一般方法	(153)
2. 应用实例	(153)
3.4 自选内容的动态显示	(154)
1. 显示内容的横向移动	(155)
2. 显示内容的纵向移动	(155)
3. 显示内容的横纵向任意移动	(157)
4. 移动显示内容的叠合	(159)
5. 应用实例	(161)
第四章 图形绘制	(165)
4.1 打印 (TEXT) 状态的语句与命令	(165)
1. 按格移笔 (LCURSOR)	(165)
2. 按格移笔打印 (LPRINT TAB)	(166)
4.2 绘图 (GRAPH) 状态的语句与命令	(170)
1. 绘图移笔 (GLCURSOR)	(171)
2. 画线 (LINE)	(172)
3. 置原点 (SORGN)	(174)
4. 置原点画线 (RLINE)	(175)
5. 字体旋转 (ROTATE)	(176)
6. 功能试验 (TEST)	(177)
4.3 应用实例	(177)
1. 图案的打印	(178)
2. 几何图形的绘制	(181)
3. 图形的变换	(186)
4. 按座标点序列绘图	(192)
5. 函数曲线和数据曲线的绘制	(201)
6. 空间曲面的绘制	(208)
7. 文字与公式的书写	(210)
8. 指定格式表格的打印或绘制	(216)
第五章 磁带文件的存取及其应用	(225)
5.1 一般介绍	(225)
1. 磁带文件——程序文件与数据文件	(225)
2. 磁带机的使用	(226)
5.2 程序文件	(228)
1. 程序文件的建立与CSAVE命令	(228)
2. 程序文件的检验与CLOAD? 命令	(228)

3. 程序文件的调用与CLOAD命令	(229)
4. 程序文件的衔接与MERGE命令	(229)
5.3 数据文件	(233)
1. 数据文件的建立与PRINT #语句 (命令)	(233)
2. 数据文件的调用与INPUT #语句 (命令)	(234)
5.4 程序的链接与CHAIN语句	(236)
5.5 两台磁带机的同时使用	(237)
1. 第一种联接方式	(237)
2. 第二种联结方式	(237)
3. 1号磁带机的使用方法	(238)
5.6 应用实例	(238)
1. PC-1500公用服务软件	(238)
2. 拉格朗日内插程序文件 (人机对话)	(241)
3. 定步长辛普生积分程序文件 (子程序)	(245)
4. 坐标点序列数据文件	(248)
(271) 5. 数据查询磁带文件	(248)
附 录	(253)
附录一 语法词汇表	(253)
附录二 PC-1500机ASCII字符编码表 (十六进制)	(256)
附录三 启动程序的三种主要方法的比较	(257)
附录四 最小BASIC语句摘要及框图表示	(257)
附录五 习题参考解答	(259)

第一章

扩充 BASIC 语言基本词法*

1.1 基本符号

BASIC 语言与大多数其它高级语言一样有一套特定的基本符号,用BASIC语言书写程序只能使用这些基本符号。

BASIC-1500的基本符号包括:

1. 二十六个大写拉丁字母: A B C.....Z

2. 二十六个小写拉丁字母: a b c.....z

3. 十个数字和 π : 0 1 2.....9 π

4. 其它符号:

+ - * / ^ \ ' ~

> < =

. , ; : " ()

! # \$ % & ? @ Δ

这里的 * 代表乘号, Δ 代表空格。各符号的意义和作用将在后面陆续介绍。

1.2 语法词汇

由基本符号可以构成一些具有特定意义的语法词汇,比如AND、SIN、LPRINT、RUN等等。它们有的在BASIC语言中做运算符使用,有的用来构成函数、语句或命令。一个完整的BASIC-1500语法词汇表列于附录一中。

这里所谓的语法词汇是从语言角度讲的。它们在本质上是一些BASIC指令,用于指示计算机进行指定的操作或计算。BASIC-1500的基本指令共有98条,其中由主机16KROM所支持的有78条,由CE-1500打印台的8K ROM所支持的有20条。这些基本指令输入内存时,都被翻译成占用2个字节的机器代码存放。每种指令对应的机器代码列入附录一中。有少数几种基本指令可构成复合指令,因此BASIC-1500语法词汇的数

*对于BASIC语言的基础知识已有初步了解的读者,可从本章开始依次阅读。初次接触BASIC语言的读者,在阅读本章程序时可参阅附录四(最小BASIC句法摘要)。

目要略多于基本指令数。

对这些语法词汇，我们在后面将逐一介绍其语法功能。

1.3 常 量

BASIC中的量根据其值的类型可分为三类：数、逻辑值和字符串。对于每一种类型的量，又可以根据其值是否可变而分为常量和变量（见表1-1）。

表 1-1 BASIC-1500中量的分类

类 型	不 可 变	可 变
数	数常量（简称数）	数变量（简称变量）
字符串	字符串常量（简称字符串）	字符串变量
逻辑值	逻辑值常量（简称逻辑值）	逻辑值变量

1. 数常量

数常量简称数，可采用不同的进制。

（1）十进制

所谓十进制就是使用十个不同的数符，逢十进一的数制。这十个数符是0，1，2，3，4，……，9。零用0表示，零加一得一，用1表示。一加一得二，用2表示。……八加一得九，用9表示。九加一得十，但此时十种数符已全部用完，故进位用10来表示。于是得到一个二位数：十位为1，个位为0。十再加一得十一，用11表示。11也是二位数，其中十位与个位皆为1，即 $11=10+1=1\times 10^1+1\times 10^0$ 。与此类似，132是三位数，其中百位、十位与个位分别为1、3、2。即：

$$132=1\times 10^2+3\times 10^1+2\times 10^0$$

一般情况下，一个十进制数可以表示为 $a_n a_{n-1} \dots a_2 a_1 a_0$ ， $a_{-1} a_{-2} \dots a_{-m}$ 。这是一个 $(n+m+1)$ 位数， n 和 m 为非负整数，式中的 a_i 代表十进制的基本符号（0、1、2……9），此数的值为：

$$\begin{aligned}
 & a_n a_{n-1} \dots a_2 a_1 a_0 . a_{-1} a_{-2} \dots a_{-m} \\
 & = a_n \times 10^n + a_{n-1} \times 10^{n-1} + \dots + a_2 \times 10^2 \\
 & \quad + a_1 \times 10^1 + a_0 \times 10^0 + a_{-1} \times 10^{-1} + a_{-2} \times 10^{-2} \\
 & \quad + \dots + a_{-m} \times 10^{-m} .
 \end{aligned} \tag{1}$$

例1.1 请将2345.67用式(1)展开。

解：对本数而言 $n=3$ ， $m=2$ 。

$$2345.67 = 2 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 5 \times 10^0 \\ + 6 \times 10^{-1} + 7 \times 10^{-2}.$$

由于人有十个手指，刚好可以与十种不同数符相对应，所以十进制自古以来为人们所熟悉，也使用得最为广泛。

(2) 二进制

数在电子计算机内是通过器件的不同物理状态来体现的，计算机使用十进制，就需要有与十种数符相应的十种不同物理状态。具有十种稳定物理状态的器件结构复杂，使用起来既不经济也不方便。相反，具有两种不同稳定状态的现象和器件却大量存在着，如开关可“开”也可“关”，线路可“通”也可“断”，磁性元件可沿两个不同方向（顺时针和逆时针）磁化等等。因此，电子计算机内部使用的是具有二个数符（与二个物理状态相应）的数制，即二进制。

二进制中的二个数符是0和1。零用0表示。零加一得一，用1表示。一加一得二，此时二种数符已用完，故进位用10表示，所以二进制是“逢二进一”，它的10表示二，也就是说，二进制的10等于十进制的2。为了避免发生混淆，常在二进制数的最后加注(2)，即二进制的10记为10(2)*，这里的二位为1，个位为0。

10(2)加1得11(2)（等于3）。11(2)再加1得100(2)（等于4），这是因为被加数11(2)的个位数1与加数1相加进位到二位，而进到二位的1又与被加数11(2)中的二位数1相加并进位到四位（这类似于十进制中的 $99+1=100$ ）。100(2)是四位为1，二位为0，个位为0的数，其值为4，参见（图1-1）。

十进制	二进制
$\begin{array}{r} 99 \\ +) \quad 1 \\ \hline 100 \end{array}$	$\begin{array}{r} 11(2) \\ +) \quad 1(2) \\ \hline 100(2) \end{array}$
$\underbrace{\quad\quad}_{\text{百十个}}$	$\underbrace{\quad\quad}_{\text{四二个}}$
$\underbrace{\quad\quad}_{\text{位位位}}$	$\underbrace{\quad\quad}_{\text{位位位}}$

图1-1 十进制与二进制的进位

表1-2的第一、二栏列出了与十进制数0-16相对应的二进制数。

对二进制数，一般可写为

$$a_n a_{n-1} \cdots a_2 a_1 a_0. a_{-1} a_{-2} \cdots a_{-m} (2) \\ = a_n \times 2^n + a_{n-1} \times 2^{n-1} + \cdots + a_2 \times 2^2 \\ + a_1 \times 2^1 + a_0 \times 2^0 + a_{-1} \times 2^{-1} \\ + a_{-2} \times 2^{-2} + \cdots + a_{-m} \times 2^{-m} \quad (2)$$

*也可以表示为(10)₂或10B，其中的B是Binary Number的简写，其意为“二进制数”。

表 1-2 0-16 的十进制、二进制和十六进制

十进制	二进制	十六进制	二—十进制
0	0	0	0000
1	1	1	0001
2	10	2	0010
3	11	3	0011
4	100	4	0100
5	101	5	0101
6	110	6	0110
7	111	7	0111
8	1000	8	1000
9	1001	9	1001
10	1010	A	0001 0000
11	1011	B	0001 0001
12	1100	C	0001 0010
13	1101	D	0001 0011
14	1110	E	0001 0100
15	1111	F	0001 0101
16	10000	10	0001 0110

这里的 n 、 m 仍为非负整数，但 a_i 是二进制数符，即只能是 0 或 1。利用式 (2) 可将任意二进制数化为十进制数。

例 1.2 请用式 (2) 将二进制数 $1101101.101(2)$ 化为十进制数。

解：对本数而言， $n=6$ ， $m=3$ 。

$$\begin{aligned}
 &1101101.101 \\
 &= 2^6 + 2^5 + 2^3 + 2^2 + 2^0 + 2^{-1} + 2^{-3} \\
 &= 109.625.
 \end{aligned}$$

(3) 十六进制

十六进制数也是本书后面经常要用到的。所谓十六进制是使用十六种数符，逢十六进一的数制。它的十六个数符分别是：0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F。这里的 A, B, ……，F 分别代表“十”，“十一”，……，“十五”。在十六进制中“十六”就再用单独数符表示，而是表示为 $10(16)^*$ ，即由个位为 0，十六位为 1 的两位数表示（参见表 1-2 第三栏）。

对于十六进制数，一般可写为

$$a_n a_{n-1} \cdots a_2 a_1 a_0 . a_{-1} a_{-2} \cdots a_{-m} (16)$$

*也可以表示为 $(10)_{16}$ 或 $10H$ ，其中 H 是 Hexadecimal Number 之简写，其意为“十六进制数”。

$$\begin{aligned}
 &= a_n \times 16^n + a_{n-1} \times 16^{n-1} + \dots \\
 &\quad + a_1 \times 16^1 + a_0 \times 16^0 + a_{-1} \times 16^{-1} \\
 &\quad + a_{-2} \times 16^{-2} + \dots + a_{-m} \times 16^{-m}.
 \end{aligned} \tag{3}$$

用此式可将任意十六进制数转换为十进制数。

例1.3 请用式(3)将6D.A(16)转换为十进制数。

解: 6D.A(16)

$$\begin{aligned}
 &= 6 \times 16^1 + 13 + 10 \times 16^{-1} \\
 &= 96 + 13 + 0.625 = 109.625.
 \end{aligned}$$

由于二进制是逢二进一的, 所以一个数目不大的数字用二进制表示就需要很多位, 书写和表示都不方便。为了避免这一缺点, 人们常常把四位二进制合并在一起用一位十六进制数表示, 也就是通过十六进制表示二进制。

例1.4 请将1101101.101表示成十六进制数。

解:

①自小数点开始分别向左和向右将每四位划归为一组(不足四位者补零)。

②参考表1-2第二、三栏中的对照表, 将每四位二进制用一位十六进制表示(见图1-2)。

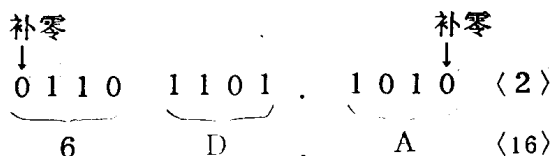


图1-2 二进制数转换为十六进制数

③ 由此可知

$$1101101.101(2) = 6D.A(16)$$

对任一个二进制数, 可以利用上述方法写出相应的十六进制数。反之, 对任一个十六进制数也可以写出与之相应的二进制数。

在计算机内的机器代码是二进制数, 为方便起见, 人们常常用十六进制书写。比如前面提到的PC-1500指令的机器代码, 在附录一中就是用十六进制书写的。

例1.5 在附录一中查得ABS和RMT ON(打开遥控)的机器代码, 其十六进制表示分别为F170和E7A9F19C, 试求它们的二进制代码。

解: ① F170(16)

$$\begin{aligned}
 &= \underbrace{1111}_{(F)} \underbrace{0001}_{(1)} \underbrace{0111}_{(7)} \underbrace{0000}_{(0)}(2).
 \end{aligned}$$

② E7A9F19C(16)

$$\begin{aligned}
 &= \underbrace{1110}_{(E)} \underbrace{0111}_{(7)} \underbrace{1010}_{(A)} \underbrace{1001}_{(9)} \underbrace{1111}_{(F)} \underbrace{0001}_{(1)} \underbrace{1001}_{(9)} \underbrace{1100}_{(C)}(2).
 \end{aligned}$$

在PC-1500机中,十六进制数的前面需冠以&,如&18即表示 $18 \langle 16 \rangle$,其值为24,&FC代表 $FC \langle 16 \rangle$,其值为252。

如果将十六进制按以上规定方法输入,则计算机可自动将其换算为十进制数并显示出来。例如:

(4) r进制

以上关于十进制、二进制和十六进制的定义可扩大到r进制。所谓r进制就是使用r种数符,逢r进一的进制。这里的r是一个大于1的正整数,每个r进制数可表示成:

$a_n a_{n-1} \dots a_2 a_1 a_0 . a_{-1} a_{-2} \dots a_{-m} \langle r \rangle$, 其值为:

输 入	显 示
& 18 ✓	24
&100 ✓	256
&ABCD ✓	43981

$$\begin{aligned}
 & a_n a_{n-1} \dots a_2 a_1 a_0 . a_{-1} a_{-2} \dots a_{-m} \langle r \rangle \\
 &= a_n \times r^n + a_{n-1} \times r^{n-1} + \dots + a_2 \times r^2 \\
 & \quad + a_1 \times r^1 + a_0 \times r^0 + a_{-1} \times r^{-1} + a_{-2} \times r^{-2} \\
 & \quad + \dots + a_{-m} \times r^{-m}.
 \end{aligned} \tag{4}$$

当 $r=10$ 、2或16时,上式即化为式(1)、(2)或(3)。这里的r称为数制的基数,如十进制的基数为10,二进制的基数为2等等。数中的每一个数位 i ($i=n, n-1, 1, 0, -1, \dots, -m$)都有一个固定的值 r^i 与之对应,这个 r^i 称为该数位的权。数中的小数点向右移动一位,数字变为原数的r倍。小数点向左移动一位,数字变为原数的 $1/r$ 。

利用式(4)可以把任何一个r进制数化为十进制数。

例1.6 请将八进制的 $625.4 \langle 8 \rangle$ 化为十进制数。

解: $625.4 \langle 8 \rangle$

$$\begin{aligned}
 &= 6 \times 8^2 + 2 \times 8^1 + 5 \times 8^0 + 4 \times 8^{-1} \\
 &= 384 + 16 + 5 + 0.5 \\
 &= 405.5
 \end{aligned}$$

如果指定一个十进制数,如何把它化为r进制呢?一般说来,这种变换要更繁杂一些,我们先考虑整数的情况。

设十进制整数为S,现求与之相应的r进制数 $a_n a_{n-1} \dots a_2 a_1 a_0 \langle r \rangle$ 。这个问题就是在已知S和r的情况下,计算 $a_n, a_{n-1}, \dots, a_2, a_1, a_0$ 的值。为此需从式(4)出发,令式(4)的值为S,将该式改写如下:

$$\begin{aligned}
 S &= a_n \times r^n + a_{n-1} \times r^{n-1} + \dots + a_2 \times r^2 + a_1 \times r^1 + a_0, \\
 &= r \times (a_n \times r^{n-1} + a_{n-1} \times r^{n-2} + \dots + a_2 \times r + a_1) + a_0, \\
 &= r \times S_1 + a_0.
 \end{aligned}$$

上式表示*: 如果以r去除S,则所得的商为 S_1 ,余数为 a_0 。这里的商 S_1 为:

*数r去除另一数S,商为 S_1 ,余数为 a_0 ;则必可写为 $S=r \times S_1 + a_0$,比如7除以3,商2余1,则可写为 $7=3 \times 2 + 1$ 。