TURBO PASCAL
程序设计

（修订本）

李连友　王黎明　编　著

河南教育出版社

前　言

计算机程序设计语言 PASCAL 是系统地体现了戴克斯特拉(E · W · Dijkstra)和霍尔(C · A · R. Hoare)定义的结构程序设计概念的第一个语言。因此它是程序设计语言发展史中的一个里程碑。PASCAL 语言是由沃斯(N · Wirth)教授于 1968 年在瑞士苏黎士联邦工业大学创立的,它是从 Algol 语言衍生而来的,但功能更强更容易使用。为了纪念最早的实用计算器的发明者、著名哲学家和数学家 Blaise Pascal 起名为 PASCAL 语言。

1971 年沃斯教授在瑞士的"ETH "杂志上发表了 PASCAL 语言，尔后，把 K · Jenson 和 N · Wirth 在"PASCAL User Manual and Report "一文中所定义的 PASCAL 语言称为标准 PASCAL。

本书主要介绍 TURBO PASCAL 的核心部分,TURBO PASCAL 是在 PC－DOS、MS－DOS、CP/M－80 和 CP/M－86 等操作系统上实现的,目前最受欢迎的 PASCAL 系统,国际上不少软件评论家称赞它实际上已成为微机的 PASCAL 标准。美国剑桥协会主席 Mark Bridger 的评论较为客观:"TURBO PASCAL 已售出 25 万份,没有其它的 PASCAL 比它更有资格成为标准,如果 Borland International 公司在 68000 机型上移植 TURBO PASCAL 成功,还有谁能说它不是微机 PASCAL 标准?"。

TURBO PASCAL 具有如下显著特点:

一、遵守标准:按沃斯的标准 PASCAL 编制的源程序无需改动就可在 TURBOPASCAL 系统下进行。

二、方便省心:TURBO PASCAL 提供了一个自完备的软件环境，一旦进入 TURBO 不需再退回操作系统,就能在菜单引导之下作所有的事,用户觉得省心,从心理上赢得了用户。

三、快　　速:TURBO 一词源于涡轮发动机(透平发动机),乃是强力,快速 之象征。TURBO PASCAL 以"三快"(编辑快，编译快，目标码执行速度快）之实符其 TURBO 之名。

四、灵　　活:体现三个方面:调用别的语言能力,被别的语言调用能力和低级接口能力。

该书是在本人编著的《TURBO PASCAL 程序设计》的基础上，再次经过王黎明同志的教学实践,增加了部分章节和例题,重新修订而成。该书具有方法的一致性,思想的一致性,实例的典型性和综合性等特点,可作为计算机及其相关专业教材,亦可供计算机工作者参考。和本书配套的有编程习题解答(软盘)。由于水平有限,书中不当之处,请读者指正。

李连友

1994 年 3 月

目　　录

第一章　计算机基础

§1.　计算机中的数和编码系统

计算机最早是做为一种计算工具出现的，所以它的最基本的功能是对数进行加工处理。数在机器中是以器件的物理状态来表示的。一个具有两种不同稳定状态且能相互转换的器件就可以表示二进制数。由于二进制数的表示简单、可靠，运算规则也非常简单，所以，目前在计算机中，数几乎全部是用二进制表示，字符用二进制编码。

一、进位计数制

“位权”和“基数”是进位计数制中的两个要素。

通常的编码方法是利用同一数码处于不同的位置代表不同数值的原理，这种和位置有关的表示法叫“位置表示法”。每个数码代表的数值等于该数码本身乘以和所在数位有关的常数值，此常数叫做“位权”，简称“权”。所允许选用的数码的个数就是计数制的“基数”。

（一）　十进制数

在日常生活中，大家最熟悉且经常使用的是十进制数，它有以下两个基本特点：

1. 具有十个不同的数字符号：0,1,2,3,4,5,6,7,8,9；

2. 逢十进一。

于是任意一个十进制数 A 都可以表示为：

$$A=A_{n-1}A_{n-2}...A_1A_0.A_{-1}A_{-2}...A_{-m}$$

A 的值为：

$$\begin{aligned}(A)_{10}&=A_{n-1}\times10^{n-1}+A_{n-2}\times10^{n-2}+...+A_1\times10^1\\&\quad+A_0\times10^0+A_{-1}\times10^{-1}+A_{-2}\times10^{-2}+...\\&\quad+A_{-m}\times10^{-m}\\&=\sum_{i=n-1}^{-m}A_i\times10^i\end{aligned}$$

其中：A_i 取 0－9 之一，n 为整数位数，m 为小数位数。

例：

$$(1234)_{10}=1\times10^3+2\times10^2+3\times10^1+4\times10^0$$

$$(0.1234)_{10}=1\times10^{-1}+2\times10^{-2}+3\times10^{-3}+4\times10^{-4}$$

$$(1234.1234)_{10}=1\times10^3+2\times10^2+3\times10^1+4\times10^0$$
$$+1\times10^{-1}+2\times10^{-2}+3\times10^{-3}+4\times10^{-4}$$

(二)　二进制数

计算机中最常使用的是二进制计数制，即二进制数，它有以下两个特点：

1. 具有两个不同的数字符号 0,1；

2. 逢二进一。

于是任意一个二进制数 B 都可以表示为：

$$B=B_{n-1}B_{n-2}...B_1B_0.B_{-1}B_{-2}...B_{-m}$$

B 的值为：

$$(B)_2=B_{n-1}\times2^{n-1}+B_{n-2}\times2^{n-2}+...+B_1\times2^1$$
$$+B_0\times2^0+B_{-1}\times2^{-1}+B_{-2}\times2^{-2}+...$$
$$+B_{-m}\times2^{-m}$$
$$=\sum_{i=r-1}^{-m}B_i\times2^i$$

其中：B_i 取 0,1 之一，n 为整数位数，m 为小数位数。

例：

$$(111.11)_2=1\times2^2+1\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}$$
$$=(7.75)_{10}$$
$$(1011.101)_2=1\times2^3+0\times2^2+1\times2^1+1\times2^0+1\times2^{-1}$$
$$+0\times2^{-2}+1\times2_{-3}$$
$$=(11.625)_{10}$$

(三)　八进制数

用二进制数表示大数所用数位较多，使用不太方便，常采用八进制。它有以下两个特点：

1. 具有八个不同的数字符号 0,1,2,3,4,5,6,7；

2. 逢八进一。

于是任意一个八进制数 C 都可以表示为：

$$C=C_{n-1}C_{n-2}...C_1C_0.C_{-1}C_{-2}...C_{-m}$$

C 的值为：

$$(C)_8=C_{n-1}\times8^{n-1}+C_{n-2}\times8^{n-2}+...+C_1\times8^1$$
$$+C_0\times8^0+C_{-1}\times8^{-1}+C_{-2}\times8^{-2}+...$$
$$+C_{-m}\times8^{-m}$$
$$=\sum_{i=n-1}^{-m}C_i\times8^i$$

其中：C_i 取 0—7 之一，n 为整数位数，m 为小数位数。

例：

$$(327)_8=3\times8^2+2\times8^1+7\times8^0$$
$$=(215)_{10}$$

$$(0.135)_8=1\times8^{-1}+3\times8^{-2}+5\times8^{-3}$$
$$=(0.18164)_{10}$$
$$(327.135)_8=3\times8^2+2\times8^1+7\times8^0+1\times8^{-1}+3\times8^{-2}$$
$$+5\times8^{-3}$$
$$=(215.18164)_{10}$$

（四） 十六进制数

目前，大部分微型机的字长是4的倍数，所以，广泛地使用十六进制数来表示，它有以下两个特点：

1. 具有十六个不同的数字符号 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F;
2. 逢十六进一。

于是任意一个十六进制数D都可以表示为：

$$D=D_{n-1}D_{n-2}\ldots D_1D_0.D_{-1}D_{-2}\ldots D_{-m}$$

D的值为：

$$(D)_{16}=D_{n-1}\times16^{n-1}+D_{n-2}\times16^{n-2}+\ldots+D_1\times16^1$$
$$+D_0\times16^0+D_{-1}\times16^{-1}+D_{-2}\times16^{-2}+\ldots$$
$$+D_{-m}\times16^{-m}$$
$$=\sum_{i=n-1}^{-m}D_i\times16^i$$

其中：D_i 取 0—9，A—F 之一，n 为整数位数，m 为小数位数。

例：

$$(327)_{16}=3\times16^2+2\times16^1+7\times16^0$$
$$=(807)_{10}$$
$$(3AB)_{16}=3\times16^2+A\times16^1+B\times16^0$$
$$=(939)_{10}$$
$$(0.11)_{16}=1\times16^{-1}+1\times16^{-2}$$
$$=(0.0664)_{10}$$
$$(3AB.11)_{16}=(939.0664)_{10}$$

通常在十六进制数末尾加上“H”来标记十六进制数。

例： 327H，3ABH，3AB.123H。

（五） 小结

上述几种计数制的特点可以概括为：

1. 每一种计数制都有一个固定的基数R，即具有R个不同的数字符号；
2. 逢R进一。

于是任意一种R进制数M都可以表示为以下通式：

$$M=M_{n-1}M_{n-2}\ldots M_1M_0.M_{-1}M_{-2}\ldots M_{-m}$$

M的值为：

$$(M)_R=M_{n-1}\times R^{n-1}+M_{n-2}\times R^{n-2}+\ldots+M_1\times R^1$$
$$+M_0\times R^0+M_{-1}\times R^{-1}+M_{-2}\times R^{-2}+\ldots$$

$$+M_{-m}\times R^{-m}=\sum_{i=n-1}^{-m}M_i\times R^i$$

其中：M_i 取 R 个字符之一，n 为整数部分位数，m 为小数部分位数。

当 M 为十进制时，M_i 取 0..9，R=10

当 M 为二进制时，M_i 取 0，1，R=2

当 M 为八进制时，M_i 取 0..7，R=8

当 M 为十六进制时，M_i 取 0..9，A..F，R=16

二、各进位制数之间的转换

(一).二进制数转换成十进制数

根据二进制数的定义，只要将它按权展开求值即可。

例：$(111.11)_2=1\times2^2+1\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}=(7.75)_{10}$

(二).十进制数转换成二进制数

1.十进制整数转换成二进制整数

设 M 为十进制整数

则：$(M)_{10}=K_{m-1}\times2^{m-1}+K_{m-2}\times2^{m-2}+\ldots+K_1\times2^1+K_0\times2^0$

$\qquad\quad=(K_{m-1}K_{m-2}\ldots K_1K_0)_2$

设 $(M)_{10}=(215)_{10}=k_{m-1}\times2^{m-1}+k_{m-2}\times2^{m-2}+\ldots+k_1\times2^1+k_0\times2^0$

两边同除以 2 得：

$(107)_{10}=k_{m-1}\times2^{m-2}+k_{m-2}\times2^{m-3}+\ldots+k_2\times2^1+k_1\times2^0\ldots$ 余 $k_0=1$

两边再同除以 2 得：

$(53)_{10}=k_{m-1}\times2^{m-3}+k_{m-2}\times2^{m-4}+\ldots+k_3\times2^1+k_2\times2^0\ldots$ 余 $k_1=1$

用这种方法一直继续下去，直到商为零，就可以得到 $k_{m-1},\ldots,k_0$.

例： $(M)_{10}=(215)_{10}=(11010111)_2$

转换过程：

2 | 215
2 | 107 ......余$1=K_0$
2 | 53 ......余$1=K_1$
2 | 26 ......余$1=K_2$
2 | 13 ......余$0=K_3$
2 | 6 ......余$1=K_4$
2 | 3 ......余$0=K_5$
2 | 1 ......余$1=K_6$
0 ......余$1=K_7$

$(215)_{10}=(K_7K_6K_5K_4K_3K_2K_1K_0)_2$

$=(11010111)_2$

又如：$(91)_{10}=(1011011)_2$

$(105)_{10}=(1101001)_2$

2. 十进制小数转换成二进制小数

设 M 为十进制小数

则：$(M)_{10}=K_{-1}\times 2^{-1}+K_{-2}\times 2^{-2}+\ldots+K_{-m}\times 2^{-m}$

$=(0.K_{-1}K_{-2}\ldots K_{-m})_2$

设$(M)_{10}=(0.6875)_{10}=K_{-1}\times 2^{-1}+K_{-2}\times 2^{-2}+\ldots+K_{-m}\times 2^{-m}$

两边同乘以 2 得：

$0.6875\times 2=1.375$

$(1.375)_{10}=K_{-1}+(K_{-2}\times 2^{-1}+K_{-3}\times 2^{-2}+\ldots+K_{-m}\times 2^{-(m-1)})\ldots K_{-1}=1$

$(0.375)_{10}=K_{-2}\times 2^{-1}+K_{-3}\times 2^{-2}+\ldots+K_{-m}\times 2^{-(m-1)}$

两边再同乘以 2 得：

$0.375\times 2=0.75$

$(0.75)_{10}=K_{-2}+(K_{-3}\times 2^{-1}+K_{-4}\times 2^{-2}+\ldots+K_{-m}\times 2^{-(m-2)})\ldots K_{-2}=0$

如此继续下去，可得到 $K_{-1},K_{-2},\ldots,K_{-m}$的值。

例： $(M)_{10}=(0.6875)_{10}=(0.1011)_2$

转换过程：

$$\begin{array}{r}0.6875\\ \underline{\times\quad 2}\\ 1.3750\end{array}\quad \cdots\cdots\cdots\text{整数部分为 }1=K_{-1}\text{，小数部分为 }0.375=M_1$$

$$\begin{array}{r}0.375\\ \underline{\times\quad 2}\\ 0.750\end{array}\quad \cdots\cdots\cdots\text{整数部分为 }0=K_{-2}\text{，小数部分为 }0.75=M_2$$

$$\begin{array}{r}0.75\\ \underline{\times\quad 2}\\ 1.50\end{array}\quad \cdots\cdots\cdots\text{整数部分为 }1=K_{-3}\text{，小数部分为 }0.50=M_3$$

$$\begin{array}{r}0.5\\ \underline{\times\quad 2}\\ 1.0\end{array}\quad \cdots\cdots\cdots\text{整数部分为 }1=K_{-4}\text{，小数部分为 }0=M_4$$

$(M)_{10}=(0.6875)_{10}$

$=(K_{-1}K_{-2}K_{-3}K_{-4})_2$

$=(0.1011)_2$

3. 小结

(1) 将一个十进制数转换成二进制数时，整数部分和小数部分的转换方式截然不同，要分别转换后再将两部分连接起来。

例： $(13.13)_{10}=(1101.001)_2$

(2) 一个二进制小数能够完全准确地转换成十进制小数。但是，一个十进制小数不

一定能完全准确地转换成二进制小数，只能根据精确度要求取相应位数得到一个近似值。

例：$(0.27)_{10} \approx (0.01001)_2$

$(0.1)_{10} \approx (0.0001100110011)_2$

（三） 任意进制数与十进制数之间的转换

1.任意进制数转换为十进制数时，将该数按权展开即可。

2.十进制数转换为任意进制数时，都要将十进制数分成整数部分和小数部分分别转换，转换过程与上面介绍的十进制向二进制转换过程相同，只是把其中的除数 2(整数部分转换过程中)和乘数 2(小数部分转换过程中)换成相应的进位制数的基数即可。

例：十进制数转换成八进制数

$(835.6875)_{10} = (1503.54)_8$

转换过程：

(1)整数部分：

8 | 835

8 | 104 ……余3

8 | 13 ……余0

8 | 1 ……余5

0 ……余1

$(835)_{10} = (1503)_8$

(2)小数部分：

$$\begin{array}{r} 0.6875 \\ \times \quad 8 \\ \hline 5.5000 \end{array}$$
……整数部分为 5

$$\begin{array}{r} 0.5 \\ \times \quad 8 \\ \hline 4.0 \end{array}$$
……整数部分为 4

$(0.6875)_{10} = (0.54)_8$

所以有：$(835.6875)_{10} = (1503.54)_8$

又如：$(835.6875)_{10} = (343.B)_{16}$

请读者自己写出转换过程。

（四） 八进制数与二进制数之间的转换

1.设 M 是一个十进制整数

则：

$$(1)(M)_{10} = L_{n-1} \times 8^{n-1} + L_{n-2} \times 8^{n-2} + \ldots + L_1 \times 8^1 + L_0 \times 8^0$$

$$(2)(M)_{10} = K_{m-1} \times 2^{m-1} + K_{m-2} \times 2^{m-2} + \ldots + K_1 \times 2^1 + K_0 \times 2^0$$

于是：$(L_{n-1}L_{n-2}\ldots L_1L_0)_8 = (K_{m-1}K_{m-2}\ldots K_1K_0)_2$

那么 L_i 与 K_j 之间有什么关系呢？

将(1)式右部除以 8 得：

(3) $L_{n-1}\times 8^{n-2}+L_{n-2}\times 8^{n-3}+\ldots+L_1\times 8^0$

........余数 L_0

将(2)式右部除以 8 得：

(4) $K_{m-1}\times 2^{m-4}+L_{m-2}\times 2^{m-5}+\ldots+K_3$

........余数 $K_2\times 2^2+K_1\times 2^1+K_0\times 2^0$

于是： $(L_0)_8=(K_2K_1K_0)_2$

将(3)，(4)式分别再除以 8 得：

$(L_1)_8=(K_5K_4K_3)_2$

如此继续下去可以得如下关系：

$(L_i)_8=(K_{3i+2}K_{3i+1}K_{3i})_2 \quad i\leqslant n-2$

当 $m-1=3n-3$ 时：$(L_{n-1})_8=(K_{m-1}K_{m-2}K_{m-3})_2$

当 $m-1=3n-4$ 时：$(L_{n-1})_8=(0K_{m-1}K_{m-2})_2$

当 $m-1=3n-5$ 时：$(L_{n-1})_8=(00K_{m-1})_2$

2. 设 M 是一个十进制小数

则：

(1) $(M)_{10}=L_{-1}\times 8^{-1}+L_{-2}\times 8^{-2}+\ldots+L_{-m}\times 8^{-m}$

(2) $(M)_{10}=K_{-1}\times 2^{-1}+K_{-2}\times 2^{-2}+\ldots+K_{-n}\times 2^{-n}$

于是：$(L_{-1}L_{-2}\ldots L_{-m})_8=(K_{-1}K_{-2}\ldots K_{-n})_2$

那么 L_i 与 K_j 之间有什么关系呢？

将(1)式右部乘以 8 得：

(3) $L_{-1}+L_{-2}\times 8^{-1}+\ldots+L_{-m}\times 8^{-m+1}$

........整数 L_{-1}

将(2)式右部乘以 8 得：

(4) $K_{-1}\times 2^2+K_{-2}\times 2^1+\ldots+K_{-n}\times 2^{-n+3}$

........整数 $K_{-1}\times 2^2+K_{-2}\times 2^1+K_{-3}\times 2^0$

于是：$(L_{-1})_8=(K_{-1}K_{-2}K_{-3})_2$

将(3)，(4)式分别再乘以 8 得：

$(L_{-2})_8=(K_{-4}K_{-5}K_{-6})_2$

如此继续下去可以得如下关系：

$(L_{-i})_8=(K_{-3i+2}K_{-3i+1}K_{-3i})_2 \qquad i\leqslant m-1$

当 $n=3m$ 时：$(L_{-m})_8=(K_{-3m+2}K_{-3m+1}K_{-3m})_2$

当 $n=3m-1$ 时：$(L_{-m})_8=(K_{-3m+2}K_{-3m+1}0)_2$

当 $n=3m-2$ 时：$(L_{-m})_8=(K_{-3m+2}00)_2$

3. 结论

一位八进制数和三位二进制数对应，整数从右到左，小数从左到右。二进制数位数不够时在末端补 0。

例 1. 八进制数转换成二进制数

$(563)_8=(101110011)_2$

转换过程：

5　6　3

101　110　011

$(0.764)_8=(0.111110100)_2$

转换过程：

0.7　6　4

111　110　100

例 2. 二进制数转换成八进制数

$(11101110011)_2=(3563)_8$

转换过程：

011　101　110　011

3　5　6　3

$(0.10101)_2=(0.52)_8$

转换过程：

0.101　010

0.5　2

（五）　十六进制数与二进制数之间的转换

十六进制数与二进制数之间的转换同八进制数与二进制数之间的转换原理完全相同，只是一位十六进制数对应四位二进制数。

例 1：十六进制数转换成二进制数

$(3AB)_{16}=(001110101011)_2$

转换过程：

3　A　B

0011　1010　1011

例 2. 二进制数转换成十六进制数

$(1101111100011.100101111)_2=(1BE3.978)_{16}$转换过程：

```
0001  1011  1110  0011  1001  0111  1000
  1     B     E     3.    9     7     8
```

（六） 小结

由上面介绍可以得到下图：

$$
\begin{tikzcd}
2 \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & 8 \arrow[d, leftrightarrow] & \\
16 \arrow[r, leftrightarrow] & 10 \arrow[r, leftrightarrow] & 2
\end{tikzcd}
$$

也就是说，除八进制和十六进制之间转换要通过二进制之外，其它进制之间都可以直接相互转换。

三、二进制编码

计算机不仅能处理数字，而且能够处理字符、符号。那么，在计算机中用于表示这些数字、字符、符号的若干位二进制数就是二进制编码。

（一）二进制编码的十进制数

一位十进制数用四位二进制编码来表示，即：

0～9 表示成 0000～1001。BCD 码就是这种表示方法的一种。

BCD 码有十个不同的数字串，且它是逢十进一的，所以它是十进制数。但它的每一位是用四位二进制数来表示的，故称为二进制编码的十进制数(BCD—Binary Coded Decimal)。

例：$(0100100101111000.000101001001)_{BCD}=(4978.149)_{10}$

由 BCD 码的定义可知，十进制与 BCD 码之间的相互转换非常直接。但 BCD 码与二进制之间的转换是不直接的，必须先把 BCD 码转换成十进制数，然后再转换成二进制数，反之亦然。

（二）字母与字符的编码

字母与字符的编码有多种方式，目前在微型机中最普遍采用的是 ASCII(Amerian Standard Code for Information Interchange 美国标准信息交换码)码。ASCII 码是 7 位二进制编码，故可表示 128 个字符。其中：

0—9 的 ASCII 码为：30H—39H，

大写字母 A—Z 的 ASCII 码为 41H—5AH。

四、二进制数的运算

在二进制数的运算中，加法、减法是最基本的运算，乘法、除法等其它数值运算可以通过加减法来实现。

(一)二进制加法

二进制加法的运算规则为：

0+0=0

1+0=1

1+1=0　进位 1

1+1+1=1　进位 1

例：1101+1011=11000

运算过程如下：

进　位		1111
被加数		1101
加　数	+	1011
和		11000

可见，两个二进制数相加时，每一位有三个数，即相加的两个数和低位的进位参加运算，用二进制的加法规则得到本位和以及向高位的进位。

(二) 二进制减法

二进制减法的运算规则为：

0－0=0

1－0=1

1－1=0

0－1=1　　有借位

例：　11000100－001100101=10011111

运算过程如下：

借　位		111111
被减数		11000100
减　数	－	00100101
差		10011111

与加法类似，每一位有三个数，本位的被减数、减数和低位的借位参加运算。

(三)二进制乘法

二进制乘法的运算规则为：

0×0=0

0×1=0

1×0=0

1×1=1

二进制乘法与十进制乘法的计算过程类似。

例：　1111×1101=11000011

运算过程如下：

被乘数　　　1111
乘　数 ×　　1011
　　　　　　1111
　　　　　 1111
　　　　　0000
＋　　　1111
　　　10100101

由于以上做法重复性差，不便于在机器中实现，为此不妨把方法变一下，即用加法和移位来实现乘法。

1. 被乘数左移的方法

我们通过例子来说明这一方法

	乘数	被乘数	部分积
	1101	1111	0000
(1)乘数最低位为 1，把被乘数加到部分积上，然后把被乘数左移。		11110	＋ 1111 1111
(2)乘数为 0，不加被乘数，被乘数左移。		111100	
(3)乘数当前位为 1，把被乘数加到部分积上，然后把被乘数左移。		1111000	1111 ＋ 111100 1001011
(4)乘数当前位为 1，把被乘数加到部分积上，得最后乘积。			1001011 ＋ 1111000 11000011

从上例中可以看到，两个 n 位数相乘，乘积为 2n 位，需要 2n 个加法器。

2. 部分积右移的方法

我们还可以通过部分积右移的方法来实现乘法，我们通过例子来说明这一方法

	乘数	被乘数	部分积	
	1101	1111	0000	1101
(1)乘数最低位为 1，把被乘数加到部分积上，然后把部分积右移。			＋ 1111 1111	1101
(2)乘数当前位为 0，不加被乘数，部分积右移。			0111	1110
(3)乘数当前位为 1，把被乘数			0011	1111

加到部分积上，然后把部分积右移。

(4)乘数当前位为 1，把被乘数加到部分积上，得最后乘积，乘积右移。

```
 +  1111 |
---------|-----
   10010 | 1111
    1001 | 0111
 +  1111 |
---------|-----
   11000 | 0111
    1100 | 0011
```

(四) 二进制除法

二进制除法与十进制除法计算过程类似。

例： 100011 ÷ 101=111

计算过程如下：

```
        000111
       ________
101    1 00011
         101
       ________
         0111
         101
       ________
          101
          101
       ________
            0
```

§2. 数的机内表示

一、机器数与真值

前面提到了二进制数，没有提到符号问题，故是一种无符号数的表示。这种表示方法，机器的全部有效位均用于表示数的大小，无符号数相当于数的绝对值的大小。

然而，数应该是有符号的，在机器内应如何表示呢？通常一个数的最高位为符号位，即若字长为八位，则 D_7 为符号位，D_6-D_0 为数字位，且符号位为 1 表示负，符号位为 0 表示正。

例： $(01001010)_2=+74$

$(11001010)_2=-74$

连同符号位在一起作为一个数，就称为机器数，而它的数值(为正数或负数)称为机器数的真值。为了运算方便，在机器中数有许多种表示方法。