

## 计算机中数制转换方法

张磊,殷世民,程家兴

(安徽大学 计算机学院,安徽 合肥 230039)

**摘要:**数制转换是应用电子技术、微机技术的一个基本知识和技能。计算机中数制的转换方法一直采用传统经典的“除基取余法”和“乘基取整法”,来分别计算数的整数部分和小数部分。文中介绍了传统的数制转换方法,同时实现了它的并行算法。还给出了更简单易学实用的转换方法,及其相应的算法。

**关键词:**数制转换;位权定位法;二进制;并行算法

**中图分类号:**TP301.6

**文献标识码:**A

**文章编号:**1673-629X(2006)11-0106-03

## The Method of Numbering System Conversion in Computer

ZHANG Lei, YIN Shi-min, CHENG Jia-xing

(Computer School, Anhui University, Hefei 230039, China)

**Abstract:** Numbering system conversion is a basic knowledge and technology of the use of electronic technology and computer technology. Traditional divided by the base for the remaining few and multiplied by the base to choose integer, each of which respectively calculates integer part and decimal part, are being applied in the method of numbering system conversion in computer. In this text, traditional numbering system conversion is introduced, meanwhile its parallel algorithm is achieved. The easier and more practical method of conversion and its corresponding algorithm is introduced too.

**Key words:** numbering system conversion; positioning method of the value of right; binary; parallel algorithm

## 0 引言

众所周知,所有信息在计算机中是以二进制形式存储的,人们习惯用十进制记数,在研究问题或讨论解题的过程中,总是用十进制数来考虑和书写。当考虑成熟后,要把问题变成计算机能够“看得懂”的形式时,就得把问题中的所有十进制数转换成二进制形式。最终计算机必须把二进制的结果转换成十进制数,显示给用户。

传统经典的转换方法用两种不同形式实现:用“除基取余法”转换十进制数整数;用“乘基取整法”转换十进制数小数。但它有一定的缺陷。文中给出了一种新的转换方法,更简单易学,转换不易出错。

## 1 传统数制转换方法

## 1.1 十进制转换为二进制的原理

用  $D, d$  分别表示十进制数的整数和小数部分。用  $B, b$  分别表示二进制数的整数和小数部分。则下式成立: $D, d = B, b$ 。

任何一个二进制数都可以按权展开,如下式:

$$D, d = B_n 2^n + B_{n-1} 2^{n-1} + \dots + B_3 2^3 + B_2 2^2 + B_1 2^1 + B_0 2^0 + b_{-1} 2^{-1} + b_{-2} 2^{-2} + \dots \quad (1)$$

收稿日期:2006-03-01

作者简介:张磊(1979-),男,安徽合肥人,硕士研究生,研究方向为优化方法;程家兴,教授,博导,研究方向智能计算与优化方法等。

下面推导十进制数转换为二进制数的方法。整数部分和小数部分,分开推导,整数部分推导如下:

二进制的整数部分: $B_n 2^n + B_{n-1} 2^{n-1} + \dots + B_3 2^3 + B_2 2^2 + B_1 2^1 + B_0 2^0$  除以 2, 可用下式表示:

$$\frac{B_n 2^n + B_{n-1} 2^{n-1} + \dots + B_3 2^3 + B_2 2^2 + B_1 2^1 + B_0 2^0}{2} =$$

$$B_n 2^{n-1} + B_{n-1} 2^{n-2} + \dots + B_3 2^2 + B_2 2^1 + B_1 2^0 + \frac{B_0}{2}$$

其中商数: $B_n 2^{n-1} + B_{n-1} 2^{n-2} + \dots + B_3 2^2 + B_2 2^1 + B_1 2^0$ , 余数: $B_0$  为 0 或 1。

再把上式的商数: $B_n 2^{n-1} + B_{n-1} 2^{n-2} + \dots + B_3 2^2 + B_2 2^1 + B_1 2^0$  除以 2, 如下式:

$$\frac{B_n 2^{n-1} + B_{n-1} 2^{n-2} + \dots + B_3 2^2 + B_2 2^1 + B_1 2^0}{2} =$$

$$B_n 2^{n-2} + B_{n-1} 2^{n-3} + \dots + B_3 2^2 + B_2 2^1 + B_1 2^0 + \frac{B_1}{2}$$

其中商数: $B_n 2^{n-2} + B_{n-1} 2^{n-3} + \dots + B_3 2^2 + B_2 2^1 + B_1 2^0$ , 余数: $B_1$  为 0 或 1。

按以上方法循环下去,直到商数等于 0 为止。此时余数  $B_n$  已经求出。

此时十进制数整数部分  $D$  已转换为二进制数整数  $B$ , 即  $D = B = B_n B_{n-1} B_{n-2} \dots B_3 B_2 B_1 B_0$ 。

下面推导十进制数小数部分转换为二进制数小数:

二进制数的小数部分: $b_{-1} 2^{-1} + b_{-2} 2^{-2} + b_{-3} 2^{-3} + \dots$

小数部分乘 2 为:  $b_{-1}2^0 + b_{-2}2^{-1} + b_{-3}2^{-2} + \dots$ 。乘以 2 后,  $b_{-1}2^0$  已经是整数了, 为二进制数字 0 或 1。

其剩余小数部分等于:  $b_{-2}2^{-1} + b_{-3}2^{-2} + \dots$ 。将其再乘以 2 后等于:  $b_{-2}2^0 + b_{-3}2^{-1} + \dots$ 。乘以 2 后,  $b_{-2}2^0$  已经是整数了, 为二进制数字 0 或 1。

按以上方法循环作下去, 直到其剩余小数部分等于 0 为止。此时十进制数  $d$  已转换为二进制数  $b$ , 即  $d = b = b_{-1}b_{-2}b_{-3}\dots$

注意: 有的小数, 按以上方法作循环, 剩余的小数部分永远不会等于 0, 成为无限循环状态, 即无法找到一个二进制数  $b$  完全等于十进制数  $d$ , 在这种情况下只能求出近似等于  $d$  的二进制数  $b$ , 因此计算机可能产生误差。

以上十进制转换为位二进制的方法可概括为: 将十进制数分为整数和小数分别转换。整数部分除 2 倒取余, 直到 0 为止。小数部分乘 2 顺取整, 直到 0 为止。最后将二进制整数部分和小数部分用小数点连接即可<sup>[1]</sup>。

## 1.2 十进制数转换二进制数

常用转换方法是用除法式和乘法式。下面给出了十进制数 125.125 转换为二进制的例子。按照上面的转换方法, 分别对整数和小数部分进行转换。

整数部分用下面左图的除法式进行转换, 如图 1 所示。小数部分用下面右图的乘法式进行转换, 如图 2<sup>[2]</sup>所示。

除数	被除数	余数		0.125	取整
2	125	1	最低位	$\times 2$	
2	62	0		0.25	0
2	31	1		$\times 2$	
2	15	1		0.5	0
2	7	1		$\times 2$	
2	3	1		1.0	1
2	1	1	最高位	- 1	
	0	← 直到 0 为止		0	← 直到 0 为止

图 1 除基取余法

图 2 乘基取整法

由此可知:  $(125)_{10} = (1111101)_2$ ,  $(0.125)_{10} = (0.001)_2$ , 把两个部分用小数点连接起来, 就得到转换的二进制数, 结果如下:  $(125.125)_{10} = (1111101.001)_2$ 。

## 1.3 十进制转换为二进制的并行算法

由式(1)可知, 将十进制数转换为其它进制数, 实际是求多项式的系数。下面对整数部分与小数部分分别讨论。

### 1.3.1 十进制整数转换成其它进制整数的并行算法

设  $D, X$  为已知的两个整数。将  $D$  转换为  $X$  进制数, 即求以下多项式的系数。

$D = B_nX^n + B_{n-1}X^{n-1} + \dots + B_3X^3 + B_2X^2 + B_1X^1 + B_0$ , 其中:  $B_i (i = 0, 1, 2, \dots, n-1)$  为非负整数;  $B_n$  为正整数, 且满足  $B_i < X (i = 0, 1, 2, \dots, n)$ , 并有  $n = \lceil \log_X D \rceil$ 。

求解上述问题的并行算法如下<sup>[3,4]</sup>:

输入:  $D, X$

输出:  $n, B_i (i = 0, 1, 2, \dots, n)$

步骤 1: 计算  $n = \lceil \log_X D \rceil$ ;

步骤 2: 按以下顺序依次计算  $x_2, x_3, \dots, x_n$

$x \cdot x \rightarrow x_2$

$x \cdot x_2 \rightarrow x_3, \quad x_2 \cdot x_2 \rightarrow x_4,$

$x \cdot x_4 \rightarrow x_5, \quad x_2 \cdot x_4 \rightarrow x_6,$

.....

$x \cdot x_2[\log_2 n] - 1 \rightarrow x_2[\log_2 n] - 1 + 1,$

$x_2 \cdot x_2[\log_2 n] - 1 \rightarrow x_2[\log_2 n] - 1 + 2, \dots,$

$x_2[\log_2 n] - 1 \cdot x_2[\log_2 n] - 1 \rightarrow x_2[\log_2 n]$ 。

步骤 3: 在第  $i$  台处理机上计算  $[1/x_i] \rightarrow d_i, i = 1, 2, \dots, n$ ;

步骤 4: 在第  $i$  台处理机上计算  $d_i - d_{i+1}, x \rightarrow a_i, i = 0, 1, \dots, n-1; d_n \rightarrow a_n$ , 这里  $d_0 = i$ 。

### 1.3.2 十进制小数转换成其它进制的小数

设  $d$  为已知的大于零的十进制小数,  $x$  为已知的正整数。将  $d$  转换为  $x$  进制的小数, 即求以下展开式的系数:

$$d = a_{-1}x^{-1} + \dots + a_{-m}x^{-m} \quad (2)$$

其中  $a_{-j}$  为非负整数, 且满足  $a_{-j} < x (j = 0, 1, 2, \dots, m)$ 。并不是所有的十进制小数都能用有限位的  $x$  进制小数表示。因此, 式(2)并不是对所有的十进制小数都成立, 对某些十进制小数转换要根据精度要求, 此时得到的是近似值。

将式(2)两边同乘以  $x_m$ , 并令  $I = \lceil d \cdot x_m \rceil, n = m - 1, a_{-j} (j = 1, 2, 3, \dots, m-1, m)$  分别用  $a_i (i = n, n-1, \dots, 1, 0)$  为非负整数, 且满足  $a_i < x (i = 0, 1, 2, \dots, n)$ 。

上述问题的并行算法如下:

输入:  $d, x, m$

输出:  $(i = 0, 1, 2, \dots, n)$

步骤 1: 计算  $I = \lceil F \cdot x_m \rceil, n = m - 1$ ;

步骤 2 ~ 步骤 4 同 1.3.1 的算法。

## 2 “位权定位法”

### 2.1 “位权定位法”原理

位权: 以基数为底, 以该数码所在位置为指数的整数次幂。如表 1 所示。

表 1 二进制各位对应的位权

...	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$	...
...	32	16	8	4	2	1	0.5	0.25	0.125	0.0625	...

位权的特点:

(1) 位权本身是一个十进制数, 这个十进制数所对应的二进制数的最高位系数为 1, 其余各位系数均为 0。  $2^n$  的位权对应最高位系数为 1, 其余各位系数为 0 的  $n+1$  位二进制数。

(2) 相邻位的位权值之间关系是高位为低位的 2 倍。

二进制数的特点: 二进制数本身仅由 0, 1 两个数码组成。

“位权定位法”正是利用了上面的 3 个特点,具体转换方法如下<sup>[5]</sup>:

第一步:定位。

根据被转换十进制数的大小,确定一个最接近又小于被转换数的位权  $2^n$ 。可知:转换结果是个  $n+1$  位二进制数,其最高位系数为 1,其余各位系数待定。

第二步:确定系数。

① 根据已确定的位权  $2^n$ ,作减法:

被转换数 - 位权  $2^n$  = 差值。

② 依次取位权  $2^{n-1}, 2^{n-2}, 2^{n-3}, \dots$  作为减数,取上一次之差值作为被减数,不断重复确定各位系数,直到差值为 0 为止。如够减,则该位系数为 1,被减数减去减数,差值作为下次被减数;如果不够减,则该位系数为 0,保留被减数。

将被转换十进制数分为整数和小数分别转换,最后合成。转换小数部分时,权值依次为  $2^{-1}, 2^{-2}, 2^{-3}, \dots$ ,不断重复步骤 ②,位权递减到多少,要依小数精确到几位来确定。

## 2.2 算法流程图

算法流程图如图 3 所示。

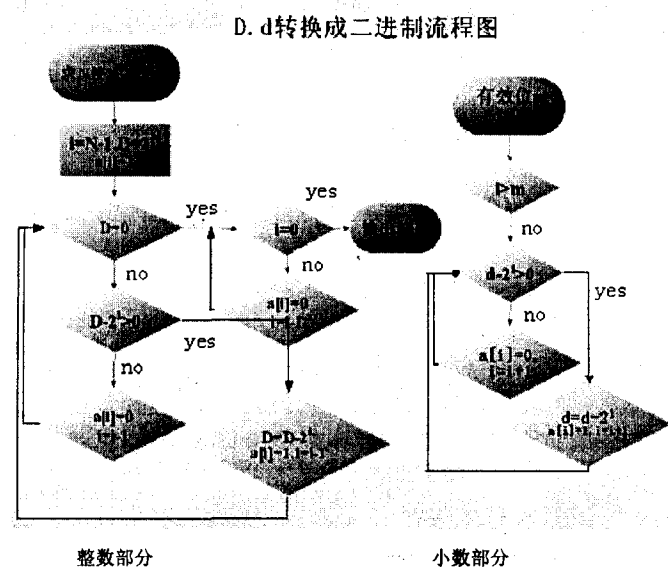


图 3 位权定位法流程图

## 2.3 “位权定位法”的应用

例如:将十进制数 125.125 转换为二进制数。

第一步:定位。

因为  $2^7 > 125.125 > 2^6$ , 所以二进制数的最高位定在第 7 位。

第二步:确定系数。

减位权直到 0 为止:  $125.125 - 64 - 32 - 16 - 8 - 4 - 1 - 0.125 = 0$ 。由理论可知,减数中出现的位权,表明够减则该位为 1,其余各位为 0。

结果为:  $(1111101.001)_2$ 。

## 3 结 论

由所举例题可知,“定位取值法”与传统经典的转换方法相比不仅简单容易,而且在转换的数值比较大时,不易出错,正确率非常高且容易掌握。

在传统的转换方法中,将十进制数分为整数部分和小数部分分别转换,再将整数和小数转换的结果中间用小数点连接即完成转换。位权定位法不用分开转换,直接减去相应的位权即可,最后将出现位权的相应位写 1,其余位补 0 即可,此数制转换的方法简便、不用连接,直接就得出结果。

## 参考文献:

- [1] 王 磊,王希雷,王 键.数制转换[J].计算机工程与应用,2003(32):95-97.
- [2] 安徽省中等职业学校计算机应用基础教材编写组.计算机应用基础[M].第 6 版.合肥:安徽教育出版社,2005:25-30.
- [3] 罗学梅,王 薇,王朝阳.数制之间相互转换的并行算法[J].山东科技大学学报:自然科学版,2003,22(4):50-51.
- [4] 陈国良.面向 21 世纪课程教材并行计算——结构、算法、编程[M].北京:高等教育出版社,2000:63-72.
- [5] 吴哲辉,曹立明,蒋昌俊.算法设计与分析[M].北京:煤炭工业出版社,1993.

(上接第 105 页)

## 参考文献:

- [1] Foster I, Kesselman C. The grid: Blueprint for a future computing infrastructure[M]. San Francisco, USA: Morgan Kaufmann Publisher, 1999.
- [2] 孙培德,胡月仙.网格计算的研究进展及应用前景[J].计算机时代,2003(1):1-5.
- [3] 都志辉,陈 渝,刘 鹏.网格计算[M].北京:清华大学出版社,2002.
- [4] 虞益诚.基于资源管理的网络技术探究[J].计算机应用与

软件,2005,22(7):69-71.

- [5] Foster I, Kesselman C, Nick J M, et al. Grid services for distributed system integration[J]. IEEE Computer, 2002, 35(6):37-46.
- [6] 罗作民,张 景,李军怀,等.网格计算及其关键技术综述[J].计算机工程与应用,2003(30):18-22.
- [7] 桂小林.基于 Internet 的元计算系统的关键技术研究[D].西安:西安交通大学,2001.
- [8] 贾明飞,董渭清,桂小林,等.网格资源管理系统模型研究[J].微电子学与计算机,2003(3):36-40.