数制转换的新型算法

魏益堂
(咸阳职业技术学院,陕西 咸阳 712000)

摘 要:按传统方法进行二进制数向十进制数转换计算,麻烦且容易出错。本文总结出了二进制数向十进制数转换时,整数部分的"8421连加法"和小数部分的"0.5依次折半求和法",使转换时的脑力劳动强度大为降低,且效率高又不容易出错。

关键词:二进制数转换;"8421连加法";"0.5依次折半求和法"

中图分类号:TP301.6 **文献标识码**:A **文章编号**:1001-7119(2012)06-0177-02

New Algorithm of the Number System Transition

WEI Yitang
(Xianyang Vocational Technical College ,Xianyang 712000,China)

Abstract:When teaching the foundation of computer science for the freshmen, it is necessary to elaborate on the knowledge of transformation from binary number to decimal number. If we follow the traditional way of transformation which is step by step, it is complex and easy to get wrong. In this paper, when I teach on this knowledge, I propose a new algorithm: "8421 continued adding" for the integral part and "0.5 sum of continued halving" for the fractional part. This algorithm can greatly reduce the workload of computation; it is more efficient and can easily avoid the faults.

Key words:transformation for binary numbers;8421 continued adding;0.5 sum of continued halving

0 引言

二进制数,形式虽简单(只有0、1两个元素),但它却是整个计算机科学的基础。没有二进制数的出现,就不会有电子计算机这个20世纪最伟大的科学技术发明。

通常人们习惯使用十进制数,而计算机内部采用二进制数来处理数值数据。因此,进行由十进制到二进制和由二进制到十进制的转换是多数应用环境的实际情况,这就迫切需要一种快速编码的方法。而一般方法"按权展开相加"没有什么技巧,费功费时。特别是在进行二进制小数部分的转换时,更是生搬硬套。计算不但复杂,而且易出错。本文在教学中摸索总结出了一套简单方便的计算方法,且不容易出错。具体的计算方法是:把一个二进制数仍分成整数和小数两部分转换。

1 二进制整数部分的十进制转换

1.1 不含"0"的特殊二进制整数部分的十进制转换

本文以二进制数11111111B为例,传统的按权展开法是:$11111111B=1\times2^7+1\times2^6+1\times2^5+1\times2^4+1\times2^3+1\times2^2+1\times2^1+1\times2^0=128+64+32+16+8+4+2+1=255$

从以上的按权展开式中,很容易看出展开的多项存在一种规律:从低位到高位是一个等比数列,其公比是2。所以没有必要对每一项都要生硬笨拙地用若干个2连乘,而是按照上述的规律可直接写出每一项的值。从11111111B的低位到高位,也就是从右边到左边,每个1所代表的十进制数值依次是1、2、4、8、16、32、64、128。所以,可以按照这个数列直接写二进制数中每个1所代表的十进制数值,然后再进行求和即可完成转换

收稿日期:2012-03-30
基金项目:咸阳职业技术学院基金项目(2010KYB04)。
作者简介:魏益堂(1960-),男,陕西户县人,讲师,研究方向:计算机学科教学研究。

工作。具体作法可以用简单的数字标注法来完成。以上述二进制数11111111B为例，其数字标注法如图1所示，然后再把所标注的十进制数相加，就是该二进制数所转换的十进制数。即：1+2+4+8+16+32+64+128=255。

```
128    32    8    2
 1  1  1  1  1  1  1  1  B
    64    16    4    1
```

图 1 8421 算法

Fig.1 8421continued adding

本文将这种方法叫“8421”法。即对于4位二进制数的整数部分来说，从左至右（从高位到低位），按照它们的等比数列的规律，所代表的十进制数依次是8、4、2、1，只需要把8、4、2、1连加起来，就可以得到4位二进制数1111B所转换的十进制数值。

当然二进制数并非都是4位，还有5位、6位及更多位数的二进制数，那也不难，只需要按照其等比数列的规律依次向上写就行了。如6位二进制数111111B中由低向高第5位1的值就是由低向高第4位1的十进制值8的二倍即16，而由低向高第6位1的值就是由低向高第5位1的十进制值16的二倍即32。由此可依次写出任意位数二进制数的每位1所表示的十进制数值，然后再进行相加就可得到二进制数到十进制数的转换。

用“8421”法对二进制数到十进制数的转换，方法简便且不易出错。对于数位少的二进制数来说，完全可以用口算就能完成转换计算，对于十位以上的多数位二进制数来说，也只需要列出如图1那样的简单数值标注图，再列个简单的加法草式，就可快速准确地完成转换计算。

1.2 对含“0”的普通二进制整数部分的十进制转换

而对于普通的二进制数即含有0的二进制数来说，用“8421”法进行转换，其方法和上面相同，只是对于“0”的处理不要简单的把它当“0”来看待，而要把它当做“1”来看待（这样做的好处就是能按照二进制数转换时的等比数列规律顺利口算出“0”后边[左边]“1”的十进制数值），然后再按图1的方法标注出它的十进制数值，但在最后进行相加时不要把它加进去。例如，转换二字节二进制数1000011111111111B为十进制数，用“8421”转换步骤如下：首先对1000011111111111B列出二进制数向十进制数的转换标注图，如图2所示：

```
32768 (8192) (2048)  512  128   32    8    2
  1  0  0  0  0  1  1  1  1  1  1  1  1  1  1  B
  (16384) (4096) 1024  256   64    16   4    1
```

图 2 十进制数的转换标记图

Fig.2 Transformation from binary number to decimal number

本文把“0”位置的十进制数值圈起来，以防误加。然后把所有标出的没有加圈的十进制数值列个简单草式或口算进行相加，如式(1)所示：

```
  3 2 7 6 8
    1 0 2 4
      5 1 2
      2 5 6
        6 4
        3 2
        1 6
          8
          4
          2
+         1
-------------
  3 4 8 1 5
```

(1)

最后相加的结果34815就是二字节二进制数1000011111111111B转换的十进制数。

2 二进制小数部分的十进制转换

对于二进制小数部分的转换，基本方法仍然是按权展开相加的办法。

2.1 对不含“0”的特殊二进制小数的十进制转换

如0.1111B=$1\times2^{-1}+1\times2^{-2}+1\times2^{-3}+1\times2^{-4}$=0.5+0.25+0.125+0.0625=0.9375

对于上面的加法，正如前述，有不少人都是采用的传统计算方法，如计算1×2-4，先要算出24的值是16，然后再列草式计算1除以16得0.0625才是1×2-4的值，而要对小数部分的每一项都要用这种办法来求，最后再把所计算出来的每项的值相加，才能得出整个小数部分的转换值。这种方法虽能完成计算，但很麻烦，也容易出错。那简单易行的方法是什么呢？本文仍然从0.1111B的转换展开式(0.5+0.25+0.125+0.0625)入手分析，发现二进制小数0.1111B小数点后从左至右的第一个1的转换值是1的折半0.5，第二个1的转换值是0.5的折半0.25，第三个1的转换值是前一个1即第二个1转换值0.25的折半0.125，第四个1的转换仍是前一个1即第三个1转换值0.125的折半0.0625。这仍是一种等比数列规律。所以，本文的简便方法就是，按照其转换过程中的等比数列规律，可依次写出它们每位小数的转换值，而不必要按照传统的指数除法来计算出它们的每一项。具体方法就是对二进制小数部分来说，每一个二进制1的转换值，就是它前一个高位1十进制值的折半。本文把这种方法称作“二进制小数从高到低0.5依次折半求和法”。

为了清晰地计算，在具体计算时，本文仍然采用前面所用的方法，列出转换标注图。如要将0.111111B转

（下转第 181 页）

重构函数，a指向欲重构的数组，num1_len为被乘数的位数，offset为第几次重构或第几次相乘运算。清单2给出该函数的主要程序代码：

```
void rebuild(int* a,int num1_len,
int offset) {
  int i=offset; while(i<num1_len+offset)
{if(a[i]>=10) {a[i+1]+=a[i]/10; a[i]%=10; }
i++;} }
```

清单2 数组重构函数

List 2: The function for reconstructing array

(3)int reverse(int* a,int size)：数组翻转函数，参数意义同上，返回值为两数计算结果的有效位数。清单3给出该函数的主要程序代码：

```
int reverse(int* a,int size) {
int i=size-1,j,temp,len; while(i>=0) {  if (a [i]!=0)  {j=i;
break;}    i--;  }
len=j+1; i=0; while(i<j){
  temp=a[i];a[i]=a[j]; [j]=temp;
  i++; j--;} return len; }
```

清单3 数组翻转函数

List 3: The function for reversing array

(4) void output(int* a,int len)：结果输出函数，len的值为两数计算结果的有效位数。

本程序的输入输出情况如图5所示。

3 结束语

本算法在进行大数相乘的处理时，具有以下几个优点：利用字符串或者字符数组避开基本数据类型所能表达数据位数有限的问题；利用一个整型数组保存中间结果，并在每次相乘计算完成后进行数组重构实现进位；最终结果在数组中反序存放，输出结果前先进行数组元素翻转，这样就可以避免处理最终结果时应该在低位输出多少个0的问题；易类推出大数（包括含小数位的大数）、超精确小数（即含多位小数位、精确度非常高的数据）的乘法运算。

参考文献：

[1] 张海藩，牟永敏. 面向对象程序设计(第二版)[M]. 北京：清华大学出版社，2007：59.

（上接第 178 页）

换成十进制数，先将二进制数0.111111B写开，再按上述的“0.5依次折半法”将每位二进制数所转换成的十进制数写在该二进制数的上面或下面，然后再把十进制数值相加：0.5 + 0.25 + 0.125 + 0.0625 + 0.03125 + 0.015625=0.984375，所得的0.984375就是二进制数0.111111B所转换成的十进制数。

2.2 对含“0”的普通二进制小数的十进制转换

对于普通的二进制小数的十进制转换问题，同样采用“0.5依次折半求和法”，只不过不要把二进制数中的某位“0”随便当“0”看待，而是先把“0”当“1”来看待进行正常转换，这样做的目的是便于“0”后面的“1”能够简单轻松（实现用口算）地实施“依次折半法”进行转换。例如把二进制小数0.11001001B转换成十进制小数，同前面一样，列出转换标注图。本文把每位二进制小数按“依次折半法”转换成的十进制小数写在其上面或下面，在此过程中要注意：把二进制数中的“0”要按“1”来对待，但其所转换的十进制数要加圈，以提醒在最后求和时，加圈的十进制数不参与求和。

3 结束语

本文提出了一种新型的二进制数转换十进制数算法，主要通过整数部分的“8421连加法”和小数部分的“0.5依次折半求和法”的方法，变传统转换法的复杂、易出错且大量笔算，为简单、方便、不易出错且大量口算的转换法。该算法减少了人们的脑力劳动强度，提高了工作效率。

参考文献：

[1] 蓝苗苗.二进制数与十、八、十六进制数之间的巧妙互换[J]. 长春理工大学学报:高教版, 2007,6(2): 132-134.

[2] 逮鸿友.关于数制转换中转换位数的确定问题[J].牡丹江师范学院学报:自然科学版, 2000,4:20-21.

[3] 郑春梅. 计算机中十进制数转换二进制数的方法比较[J].中国科教创新导刊,2007,14: 58.

[4] 吴哲. 实用的十进制数到十六进制数的FORTRAN转换程序[J].计算机应用研究,1989,2: 47-48.

[5] 吴菊凤，陈雪梨.数制转换过程中小数的“有限—无限”现象[J].绍兴文理学院学报,2011,3(31): 16-19.

[6] 蒋福德.谈“权值法”攻克数制转换中的难点及其推广[J].教育与职业,2006,1(2): 104-106.

[7] 张磊，殷世民，程家兴.计算机中数制转换方法[J].计算机技术与发展,2006,11(16): 106-108.

[8] 陈清华，郑涛，陈家伟.数制转换的本质和方法[J].江西师范大学学报:自然科学版,2006,3(30):123-125.

数制转换的新型算法

作者： 魏益堂， WEI Yitang
作者单位： 咸阳职业技术学院,陕西咸阳,712000
刊名： 科技通报 ISTIC PKU
英文刊名： Bulletin of Science and Technology
年，卷(期)： 2012,28(6)

参考文献(8条)

1. 蓝苗苗 二进制数与十、八、十六进制数之间的巧妙互换 2007(02)

2. 逮鸿友 关于数制转换中转换位数的确定问题 2000

3. 郑春梅 计算机中十进制数转换二进制数的方法比较 2007

4. 吴哲 实用的十进制数到十六进制数的FORTRAN转换程序 1989

5. 吴菊凤;陈雪梨 数制转换过程中小数的"有限-无限"现象 2011(31)

6. 蒋福德 谈"权值法"攻克数制转换中的难点及其推广 2006(02)

7. 张磊;殷世民;程家兴 计算机中数制转换方法 2006(16)

8. 陈清华;郑涛;陈家伟 数制转换的本质和方法 2006(30)