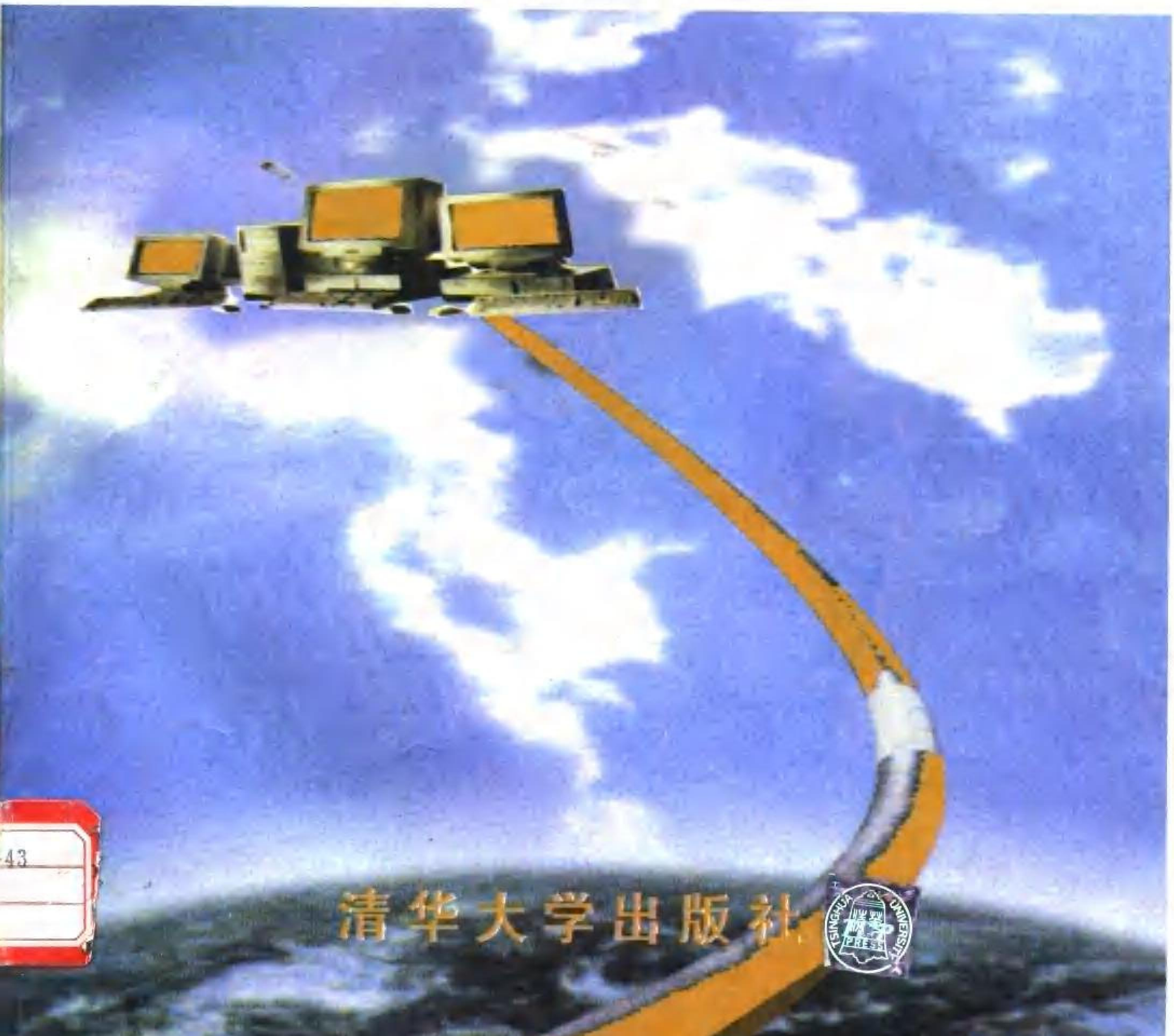
计算机专业大专系列教材

数字电路逻辑设计

李大友 主编
严化南 顾喜隆 编著



清华大学出版社

计算机专业大专系列教材

数字电路逻辑设计

李大友 主编
严化南 顾喜隆 编著

清华大学出版社

(京)新登字 158 号

内 容 提 要

全书共分 11 章，内容包括数字电路的基础知识、组合电路的分析与设计、组合集成电路、时序电路的分析与设计、集成时序电路、大规模集成电路 ROM，RAM 及 PLD、脉冲信号的产生与整形、A/D 和 D/A变换等。

本书在讲清基本理论的前提下，加强了对中、大规模集成电路的介绍和应用。

本书叙述精简扼要。可作为高等学校计算机专业大专教材、也可作为自动化类、电子类、电力类等专业的教材，也可供有关工程技术人员学习和参考。

图书在版编目(CIP)数据

数字电路逻辑设计/李大友主编；严化南，顾喜隆编著．—北京：清华大学出版社，1997.7

ISBN 7-302-02587-8

Ⅰ．数…　Ⅱ．①李…　②严…　③顾…　Ⅲ．数字电路-逻辑设计　Ⅳ．TN711.52

中国版本图书馆 CIP 数据核字(97)第 13307 号

出版者：清华大学出版社（北京清华大学校内，邮编 100084）
印刷者：昌平环球印刷厂
发行者：新华书店总店北京科技发行所
开　本：787×1092 1/16　印张：19　字数：445 千字
版　次：1997 年 9 月 第 1 版　1997 年 9 月 第 1 次印刷
书　号：ISBN 7-302-02587-8/TP・1320
印　数：0001～6000
定　价：20.00 元

序　言

为什么要组织编写这套计算机专业大专使用的教材？根据什么来组织这套教材？这套教材的特点是什么？它能起到哪些作用？这就是我们在这篇序言中，要回答，也必须回答的问题。

计算机专业大专教育发展非常迅速，它满足了社会对这个层次人材的需求。在数量上已经超过了对本科人材的需求。大专这个层次有自己的特殊性，时间只有三年，要学习的内容很多，怎样精选教学内容，就成为十分重要的问题；他们又不同于中专层次，要求既有相当坚实的理论基础，又要运用理论解决实际问题，因此，如何处理好理论和实践的关系就十分重要。

大专这个层次人材的重要性是不言而喻的。但是，在培养大专这个层次人材的过程中，突出的矛盾之一，就是缺乏合适的大专教材。目前，不是用本科教材代用，就是很难及时获得所需教材。这就是组织这套专为大专使用的教材的起因。

那么，我们组织编写这套教材以什么为依据呢？中国计算机学会教育委员会与全国高等学校计算机研究会联合推荐的《计算机学科教学计划 1993》是我们这次组织编写这套教材的主要依据。

"93 计划"所提供的指导思想和学科内容不仅适合大学本科，也适合于大专的需要。

"93 计划"明确规定了计划实施的目标：

1. 要为"计算机学科"的毕业生提供一个广泛坚实的基础；

2. 在培养人材的过程中，必须反映培养目标的差异；

3. 要为学生毕业后，进一步学习新的知识和迎接新的工作挑战，做好理论和实践上的准备；

4. 要学生能够把在校学到的知识，用到解决实际问题的过程中去。

在学科内容方面"93 计划"概括了九个科目领域。九个科目领域组成《计算机学科》的主科目。每个科目领域都有重要的理论基础、重要的抽象（实验科学）、重要的设计和实现的成就。

这九个科目领域作为教学计划的公共要求，它们是：算法与数据结构、计算机体系结构、人工智能与机器人学、数据库与信息检索、人—机通信、数值与符号计算、操作系统、程序设计语言、软件方法学和工程。

我们根据上述指导思想和学科要求精选了十三门课，作为大专用的主干教材。它们是：《数据结构》、《数字电路逻辑设计》、《计算机组成原理》、《微机原理与应用》、《微机接口技术》、《计算机网络》、《数据库原理及应用》、《操作系统基础》、《汇编语言程序设计》、《C 程序设计》、《软件工程概论》、《微机系统应用基础》和《离散数学》。

这十三门教材大体上反映了除人工智能与机器人学和数值与符号计算之外的全部要求，足以满足大专主干课程教学的需求。

这套教材我们都是聘请大专院校有丰富教学实践经验的工作在第一线的专家、教授编写。在编写过程中，充分考虑了大专的特点，在选材上贯彻少而精的原则，在处理上贯彻理论密切联系实际的原则。力求深入浅出，便于教学。并且在主要章节后均附有适量的习题。

这套教材适合于计算机专业大专生使用，也可作为非计算机专业的本科生使用。

主编　李大友

1996.6

前　言

本书是依据计算机专业大专系列教材大纲并结合近年来的教学实践编写的。讲授全书大约需要 60 学时左右。

全书共分 11 章。除介绍最基础的理论外，重点介绍中、大规模集成电路。

第 1～第 3 章是预备知识。内容包括数制与编码、逻辑代数及门电路。主要介绍二进制及其他各种进位计数制之间的相互转换、各种编码的十进制数、常见的可靠性编码、逻辑代数的基本概念和逻辑函数的化简、TTL 和 MOS 集成电路的构成和外部特性等。这些知识是进一步研究逻辑电路的组成和应用所必备的。

第 4 章是组合逻辑电路。除介绍传统的分析方法和设计方法外，重点介绍了常见的各种中规模集成电路的特性和应用。

第 5～第 8 章是时序电路。主要介绍同步时序电路、脉冲异步时序电路的分析方法和设计方法。这些方法属于经典开关理论的最小化设计，看起来内容似乎有些庞杂，但仍然是逻辑设计的基础理论。此外，还介绍了各种常见的时序集成电路。为了减少篇幅、突出重点，主要介绍外部特性和使用，对其内部结构，通常只给出原理图，而不作过多的解释。

第 9 章是存储器和可编程逻辑器件。可编程器件(PLD)是近十几年来发展起来的新型集成电路。一片 PLD 可代替几十、几百甚至上千个逻辑门，是逻辑电路的重要分支。这一章简要地说明了 PLD 的结构、原理和常用的开发软件工具，力求使读者学习后能具有实际应用的能力。

第 10 章是脉冲信号的产生与整形，第 11 章是 D/A 和 A/D 变换。这两章除介绍基本原理外，还介绍了典型的集成电路芯片。

全书采用国家标准的图形符号，并对复杂符号中的限定符作了简要说明。

本书第 1,2,5,6 章和第 11 章的 A/D 变换部分由顾喜隆编写，其余部分由严化南编写，由严化南统编。

本书承蒙本系列教材的主编李大友教授作了全面的审阅和修改，编者深表感谢。

由于作者水平有限，书中不妥乃至错误之处，请广大读者批评指正。

编　者

1996 年 11 月

目　录

第1章　数制与编码

本章是数字电路的基础，将介绍进位计数制的规则及各种进位制数之间的转换。由于数字系统只能识别二进制数，所以本章将重点阐述二进制数的表示。此外还简单介绍了几种常见的二—十进制代码及可靠性编码和字符代码。

1.1　进位计数制

十进制是我们最为熟悉的进位计数制。它之所以是十进制是因为它采用了10个有序数字符号0,1,2,3,4,5,6,7,8,9代表一位十进制数的十个不同状态，并执行逢十进一，借一当十的原则。从而看出，计数制实际上是以计数符号的个数来命名的。我们称计数符号的个数为基数。

在十进制数中，当数码处于不同位置时，所表示的位值是不同的。

如

$$\begin{array}{ccc} & 8\quad 8\quad 8 & \\ 8\times 100 & 8\times 10 & 8\times 1\end{array}$$

从右至左这个三位十进制数每位的数码都是8，但位值依次分别是100,10,1。位值又称权值或位权。

有了基数和位权的概念，对于任一个十进制数N按其位权值展开均可表示为：

$$\begin{aligned}(N)_{10} &= a_{n-1}\times 10^{n-1} + a_{n-2}\times 10^{n-2} + \cdots + a_1\times 10 + a_0\times 10^0 \\ &\quad + a_{-1}\times 10^{-1} + a_{-2}\times 10^{-2} + \cdots + a_{-m}\times 10^{-m} \\ &= \sum_{i=n-1}^{-m} a_i \times 10^i \end{aligned} \tag{1-1}$$

式中a_i为0～9中任一数码，n和m为正整数，n为整数部分的位数，m为小数部分的位数。那么，对于任意进制数，我们可以写成下式：

$$(N)_R = \sum_{i=n-1}^{-m} r_i \times R^i \tag{1-2}$$

式中r_i为任意进制中第i位的数码，n和m为正整数，n为整数部分的位数，m为小数部分的位数，R为进位基数，R^i为第i位的权值。我们称以上二式分别为十进制数和任意进制数的按权展开式。例如一个十进制数12.34可表示为：

$$(12.34)_{10} = 1\times 10^1 + 2\times 10^0 + 3\times 10^{-1} + 4\times 10^{-2}$$

综上所述，进位计数制不论进位基数如何，它们的规律是相同的：

(1) 进位基数是固定的，并且必须是大于1的正整数；

(2) R进制中的每一个数位规定可使用的数码个数为R个，其中最小的数码为0，最大的数码为$(R-1)$；各位数码a_i可取R个数码中的任何一个。

(3) R进制的任何数均可用按权展开式来表达，整数部分的位权用R的正指数幂表

示。小数部分用 R 的负指数幂表示。

本书中常用的进位计数制是十进制(Decimal)、二进制(Binary)、八进制(Octadic)、十六进制(Hexadecimal)。因此，当基数 R 为 10 时，表示十进制数也可用 $(N)_D$ 表示。同样二进制数，八进制，十六进制数可分别用 $(N)_B$，$(N)_O$，$(N)_H$ 表示，但是 $(N)_O$ 中的脚标 O 为与数字 0 相区别，常用字母 Q 表示，即 $(N)_Q$。

1.2 二进制数

按上节所讲进位计数制的规律，在式(1-2)中当 R 为 2 时，该式表示一个二进制数(Binary number)。二进制数的每个数位可用数码只有 0 和 1，且逢二进一，即每位计到两个数时就要向高位进一，同时本位变零，所以二进制数中“10”表示十进制数的 2。在二进制中不同数位的权值也不相同，其相邻各位的权值是以 2 为底的连续整数次幂，即相差一倍。因此任意一个二进制数 N 可以表示为：

$$(N)_2 = \sum_{i=n-1}^{-m} b_i \times 2^i \tag{1-3}$$

式中 b_i 只能取 0 或者 1 两个数码。

如 $(1101.01)_2 = 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 + 0\times2^{-1} + 0\times2^{-2}$

二进制数各位数值如表 1.1 所示。

表 1.1 二进制($R=2$)各位数值(2^i)

i	2^i	i	2^i	i	2^i
-1	0.5	0	1	5	32
-2	0.25	1	2	6	64
-3	0.125	2	4	7	128
-4	0.0625	3	8	8	256
-5	0.03125	4	16	9	512
⋮	⋮			⋮	⋮

注：表中各值均用十进制数表示。

二进制数的运算规则如下所示：

加 法	乘 法
$0+0=0$	$0\times0=0$
$0+1=1+0=1$	$0\times1=1\times0=0$
$1+1=10$	$1\times1=1$

下面举几个二进制整数四则运算的例子：

加　法

$1011+110=10001$

$$\begin{array}{r} 1011 \\ +\quad 110 \\ \hline 10001 \end{array}$$

减　法

$1011-110=101$

$$\begin{array}{r} 1011 \\ -\quad 110 \\ \hline 101 \end{array}$$

乘　法

$1101\times110=1001110$

$$\begin{array}{r} 1101 \\ \times\quad 110 \\ \hline 0000 \\ 1101 \\ +\ 1101 \\ \hline 1001110 \end{array}$$

除　法

$11001\div101=101$

$$\begin{array}{r} 101 \\ 101\overline{)\,11001} \\ -\ 101 \\ \hline 101 \\ -\ 101 \\ \hline 0 \end{array}$$

二进制数在计算机中是很易表示与实现的(0,1 两状态可用逻辑元件实现),且运算简单,因此被数字系统广泛采用。

1.3　八进制数和十六进制数

二进制数虽在计算机中易于实现,然而它最大的缺点是不便读写,与十进制数相比,表示同一个数时二进制数用的位数较多。为了便于说明由二进制所确定的系统,减少数位,一般在讲述和书面表达时,常采用八进制和十六进制数。

进位基数 $R=8$ 时,称为八进制。八进制数 $(N)_Q$ 的按权展开式可写成:

$$(N)_Q = q_{n-1}\times 8^{n-1} + \cdots + q_1\times 8^1 + q_0\times 8^0 + q_{-1}\times 8^{-1} + \cdots + q_{-m}\times 8^{-m}$$

即
$$(N)_Q = \sum_{i=n-1}^{-m} q_i \times 8^i \tag{1-4}$$

式中 q_i 只能取 0～7 中的某一个数码

如　$(327.6)_Q=3\times8^2+2\times8^1+7\times8^0+6\times8^{-1}$

同理 $R=16$ 时称为十六进制数。十六进制数的计数规律是逢十六进一。这 16 个计数符号是 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F。这里十进制数的 10～15 分别用 A～F 6 个英文字母表示。

十六进制数 $(N)_H$ 的按权展开式为:

$$(N)_H=h_{n-1}\times16^{n-1}+\cdots+h_1\times16^1+h_0\times16^0+h_{-1}\times16^{-1}+\cdots+h_{-m}\times16^{-m}$$

即
$$(N)_H = \sum_{i=n-1}^{-m} h_i \times 16^i \tag{1-5}$$

式中 h_i 只能取 0～F 中的某一个数码。

如　$(3A.C)_H=3\times16^1+A\times16^0+C\times16^{-1}$

表 1.2 列出了与十进制数 0～16 相对应的二进制、八进制、十六进制数。

表 1.2　与十进制数相对应的二、八、十六进制数

十进制	二进制	八进制	十六进制
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

1.4　数制转换

二进制数具有易于表示、方便运算和易于实现的优点，但使用时位数较多，人们读写很不方便；十进制数虽然是生活中最常用、最习惯的一种进位制数，但由于每位数所使用的数码较多计算机实现起来所需要的设备量大，而且不易实现。因此，常常需要把十进制数转换成二进制数；而为方便读写，以适应人们的习惯，又要进行相反的变换。因此在使用中，经常需进行不同数制之间的转换。

数制之间的转换，可归为两类：十进制数和非十进制数之间的转换；2^n 进制数之间的转换。

1.4.1　十进制数和非十进制数之间的转换

1. 非十进制数转换成十进制数

我们知道任何一个数都可用其按权展开式表示为：

$$N = \sum_{i=n-1}^{-m} a_i R^i$$

于是我们很容易将一个非十进制数转换成相应的十进制数。具体方法是：将一非十进制数按权展开成一多项和式，每项是该位数码与相应权值之积，把此多项式在十进制系统中进行计算，所得结果就是该数的十进制形式。这种方法也称多项式替代法。

例 1　将二进制数 1011.011 转换成十进制数。

$$(1011.011)_B = 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-2} + 1 \times 2^{-3}$$

$$=8 + 2 + 1 + 0.25 + 0.125$$

$$=(11.375)_D$$

例 2　将八进制数$(27.46)_Q$转换成十进制数。

$$(27.46)_Q = 2 \times 8^1 + 7 \times 8^0 + 4 \times 8^{-1} + 6 \times 8^{-2}$$

$$= 16 + 7 + 0.5 + 0.09375$$

$$= (23.59375)_D$$

2. 十进制数转换成非十进制数

十进制数转换成非十进制数时，要将其整数部分和小数部分分别转换，结果合并为目的数制形式。

(1) 整数部分的转换

整数部分的转换采用基数除法。所谓基数除法即用目的数制的基数去除十进制整数，第一次除所得余数为目的数的最低位，把得到的商再除以该基数，所得余数为目的数的次低位，依次类推，继续上面的过程，直至商为 0 时，所得余数为目的数的最高位。此法也叫除基取余法。

例　把$(53)_D$转换成二进制数。

2	53		低位
2	26	……1	↑
2	13	……0	
2	6	……1	
2	3	……0	
2	1	……1	
	0	……1	高位

$(53)_D = (110101)_B$

(2) 小数部分的转换

小数部分的转换是采用基数乘法进行的。所谓基数乘法即用该小数乘目的数制的基数，第一次乘得结果的整数部分为目的数的最高位（当然是小数部分的最高位），其小数部分再乘基数，所得结果的整数部分做为目的数的第二位，依次类推，继续上面的过程，直到小数部分为 0 或达到要求精度为止。

例　将十进制小数$(36.6875)_D$ 转换成八进制数。

整数部分和小数部分分别用基数除和基数乘法进行转换。

8	36	
8	4	……4
	0	……4

$(36)_D = (44)_Q$

0.6875		
× 8		
$\boxed{5}$.5000	……5	高位
× 8		↓
$\boxed{4}$.0	……4	低位

$(0.6875)_D = (0.54)_Q$

$\therefore (36.6875)_D=(44.54)_Q$

基数乘法也叫乘基取整法。基数乘除法适用于十进制数向任意非十进制数的转换。

值得注意的是，任何二进制小数都可以完全地转换成十进制小数，但十进制小数有时不能用二进制小数精确地表示出来，这时只能根据精度要求，求到一定的位数，近似地表示。

例 将$(0.423)_D$转换成二进制小数(保留 4 位小数)

$$
\begin{array}{rl}
0.423 & \\
\times \qquad 2 & \\
\hline
\boxed{0}.846 & \cdots\cdots 0 \\
\times \qquad 2 & \\
\hline
\boxed{1}.692 & \cdots\cdots 1 \\
\times \qquad 2 & \\
\hline
\boxed{1}.384 & \cdots\cdots 1 \\
\times \qquad 2 & \\
\hline
\boxed{0}.768 & \cdots\cdots 0 \\
\times \qquad 2 & \\
\hline
\boxed{1}.536 & \cdots\cdots 1
\end{array}
$$

$$(0.423)_D=(0.0111)_B$$

一般要保留 4 位小数，则第 5 位小数采取“零舍一入”的原则。

(3) 数制转换时小数位数的确定

设 α 进制小数有 k 位，转换成 β 进制后的某一精度需要 j 位，这时应有：

$$(0.1)_\alpha^k = (0.1)_\beta^j$$

这在十进制中应写成：

$$\left(\frac{1}{\alpha}\right)^k = \left(\frac{1}{\beta}\right)^j$$

对某式两边都取基数 α 的对数：

$$k\log\alpha\left(\frac{1}{\alpha}\right) = j\log\alpha\left(\frac{1}{\beta}\right)$$

故有

$$k\log\alpha(\alpha) = j\log\alpha(\beta)$$

$$k = j\log\alpha(\beta)$$

利用等式

$$\log\alpha(\beta) = \frac{\log_{10}(\beta)}{\log_{10}(\alpha)}$$

可以得到

$$k=j\frac{\log_{10}(\beta)}{\log_{10}(\alpha)} \quad 或 \quad j=k\frac{\log_{10}(\alpha)}{\log_{10}(\beta)}$$

当然这样算出的 j 一般不是一个整数，所以我们应该选择满足下列不等式的整数：

$$k\frac{\log_{10}(\alpha)}{\log_{10}(\beta)} \leqslant j \leqslant k\frac{\log_{10}(\alpha)}{\log_{10}(\beta)} + 1$$