中等专业学校教材

数字逻辑

唐德洲 编

哈尔滨工业大学出版社

数 字 逻 辑

唐德洲 编

哈尔滨工业大学出版社

JS229/30-05

内 容 简 介

本书是根据机械电子工业部中专电子类教材编审委员会计算机专业教材编审小组通过的编写大纲编写的。

本书主要内容包括：数制编码与逻辑代数、组合逻辑电路的分析与设计、同步时序电路的分析与设计、异步时序电路的分析与设计、常见时序电路及综合应用，重点阐述数字逻辑电路的基本概念、常用分析方法和设计方法。在分析常见的数字逻辑部件时，介绍相应的中规模集成器件的逻辑功能及典型应用。每章均附有小结及习题。

本书可作为中等专业学校电子计算机专业技术基础课的教材，也可供有关科技人员参考。

数 字 逻 辑

Shuzi Luoji

唐德洲 编

*

哈尔滨工业大学出版社出版

新华书店首都发行所发行

肇东粮食印刷厂印刷

*

开本 787×1092 1/16 印张 14.75 字数 337 千字

1992 年 10 月第 1 版 1996 年 8 月第 3 次印刷

印数 9 001—19 000

ISBN 7-5603-0345-5/TP・29 定价 14.00 元

出 版 说 明

根据国务院关于高等学校教材工作分工的规定，我部承担了全国高等学校、中等专业学校工科电子类专业教材的编审、出版的组织工作。由于各有关院校及参与编审工作的广大教师共同努力，有关出版社的紧密配合，从 1978 年至 1985 年，已编审、出版了两轮教材，正在陆续供给高等学校和中等专业学校教学使用。

为了使工科电子类专业教材能更好地适应“三个面向”的需要，贯彻“努力提高教材质量，逐步实现教材多样化，增加不同品种、不同层次、不同学术观点、不同风格、不同改革实验的教材”的精神，我部所属的有关七个高等学校教材编审委员会和两个中等专业学校教材编审委员会，在总结前两轮教材工作的基础上，结合教育形势的发展和教学改革的需要，制订了 1986～1990 年的“七五”(第三轮)教材编审出版规划。列入规划的教材、实验教材、教学参考书等近 400 种选题。这批教材的评选推荐和编写工作由各编委会直接组织进行。

这批教材的书稿，是从通过教学实践、师生反映较好的讲义中经院校推荐，由编审委员会(小组)评选择优产生出来的。广大编审者、各编审委员会和有关出版社为保证教材的出版和提高教材的质量，作出了不懈的努力。

限于水平和经验，这批教材的编审、出版工作还会有缺点和不足之处，希望使用教材的单位、广大教师和同学积极提出批评建议，共同为不断提高工科电子类专业教材的质量而努力。

机械电子工业部电子类专业教材办公室

前　言

本教材系按机械电子工业部的工科电子类专业教材1986～1990年编审出版规划，由中专电子类教材编审委员会计算机教材编审小组征稿、推荐出版的，责任编委范理风。

本教材由成都电子机械高等专科学校(原成都无线电机械学校)唐德洲担任主编，常州无线电工业学校凌林海担任主审。

本课程的参考学时数为90学时，其中理论课约占70学时，实验课20学时。本教材主要内容包括数制与编码、逻辑代数、组合逻辑电路的分析与设计、同步时序逻辑电路的分析与设计、异步时序逻辑电路的分析与设计以及综合应用举例。本课程的中心任务是阐明数字逻辑电路的基本概念和通用的逻辑电路设计方法。鉴于目前集成电路的飞速发展和广泛应用，在分析基本逻辑部件的同时，介绍相应的中规模集成电路的功能及典型应用。本教材中介绍的集成电路实例大都选自国产集成电路手册。

本课程是电子计算机专业的一门技术基础课。它的先导课是“计算机电路基础”，故本教材以门电路为起点，从利用外特性的角度，回顾在“计算机电路基础”课程中学过的各种门电路及各类触发器的逻辑功能，使抽象的逻辑设计落实到具体的基本元件上，对门电路和触发器的内部结构、电路组成和工作原理等内容不再涉及。本教材的重点是阐明数字逻辑电路的基本概念，培养学生分析逻辑电路和进行逻辑电路设计的能力。编写时，在文字上力求简明扼要，通俗易懂，便于自学。各章都附有小结和一定数量的习题。

在本教材的编写过程中，参加1987年8月在哈尔滨召开的“中专计算机教材编审组工作会议及教材评选会”的编委及编辑对编者给予了极大的鼓励，并提出了许多宝贵的建议。主审凌林海同志对修改稿进行了全面、细致的复审，提出了极其宝贵的意见。在此表示诚挚的感谢。

由于编者水平有限，书中难免还存在一些缺点和错误，殷切希望广大读者批评指正。

编　者

目　　录

绪 论

数字电路已在电子计算机、通讯、电视、雷达、自动控制、电子测量仪器、核物理、航天等许多领域中得到广泛的应用。集成电路的发展，特别是中、大规模和超大规模数字集成电路的发展，已经给各类电子系统，特别是电子计算机的发展，带来了深远的影响。利用数字电路构成的测量仪器，不仅具有测量精度高、测试功能强的优点，而且还能进行数据处理，实现测量的自动化、智能化。用数字电路构成的数字通讯系统，抗干扰能力强，保密性好，适用于多路远程传输，而且还能应用计算机进行信息处理和控制，实现以计算机为中心的自动交换通讯网。用数字电路构成的各式各样的数字控制装置，可以用来实现对生产过程的自动控制。现在，大规模和超大规模集成电路的发展已经能把一整台计算机制造在一块半导体材料上，这就是微型计算机和单片计算机。由于微型计算机和单片计算机的集成规模大，体积小，通用性强，价格低廉，故原来只用一般数字电路构成的数字化仪器表、数字程序控制机床和其它数字控制设备正在逐步用微型计算机和单片计算机所取代。

随着数字电路的应用领域的不断扩大，对现代科学技术、国民经济和人民的生活正在产生着越来越深刻的影响。所以，讲述上述内容的"数字逻辑"课程就成为电子类各专业，特别是电子计算机专业的一门重要的技术基础课。

1. 数字信号

在"计算机电路基础"课程中，所遇到的信号(如正弦信号等)，在时间上和数值上都是连续变化的，这种信号称为模拟信号。在数字电路中所要处理和存储的信号，在时间上是离散的，数值上是整量化了的，这种信号称为数字信号。各种模拟物理量的模拟信号，都必须转换为数字信号，才能送入数字系统中进行加工处理。在计算机和数字系统中，信息中的英文字母 A、B、C…，希腊字母 α、β、γ…，算术运算符号＋、－、×、÷等，都用数字信号表示。在数字系统中，采用最多的是只有 0、1 两种数值组成的数字信号。这种二值数值信号又称为二进制信号。

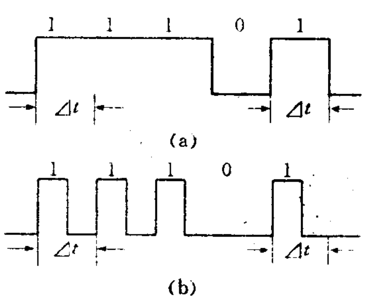

图0-1 数字信号

二值数字信号容易用电路实现，在传送和处理时不容易出错。通常，0 和 1 两个二进制数字信号是采用电位的高、低或脉冲信号的有、无两种不同状态来代表的，故数字信号分为电位型和脉冲型两种。

图 0-1(a)所示为电位型数字信号，它是用高电位表示"1"，低电位表示"0"。通常，每个 1、0 信号的持续时间 Δt 是相等的，每个 Δt 称为一拍(位)，几个连续的 1 或连续的 0，就是几拍长的高电位或低电位，图 0-1(a)所示信号就是 11101 的五拍二进制信号。图 0-1(b)所示为脉冲型数字信号，有脉冲表示"1"，无脉冲表示"0"，这样图0-1(b)所

示信号也是 11101 五拍二进制信号。

2. 数字电路和数字系统

数字电路指的是能对数字信号进行算术运算和逻辑运算的电路。所谓算术运算，即对两个或两个以上数字信号进行+、-、×、÷等一系列算术加工；所谓逻辑运算，即对数字信号进行与、或、非以及与非、或非、异或等逻辑关系的加工处理及控制。因此数字电路也常被称为“数字逻辑电路”。

对于数字逻辑电路的分析和设计要考虑两方面的问题。一方面是电路的电气性能，如基本数字集成电路(门电路、触发器等)的工作原理、静态特性和动态特性等。这在“计算机电路基础”课中已有详尽介绍，本书不再赘述。另一方面是电路的逻辑功能，即输入信号和输出信号之间的逻辑关系。为了分析这些逻辑关系，需要采取一套新的分析方法，其中包括逻辑代数、真值表、卡诺图、特性方程及状态图等。为了清楚地表达复杂数字逻辑电路的逻辑功能，通常把基本逻辑电路用不同的图形符号来表示。这种由一系列图形符号及其之间的逻辑关系所构成的电路图叫逻辑图，或称为逻辑电路。无论在数字逻辑电路的分析还是设计中，逻辑图和逻辑功能之间的关系，都是讨论的中心课题。

数字系统就是以数字逻辑电路为主，传送和加工处理数字信息的实际工程系统。数字逻辑电路的输入和输出信号都是数字信号，因此数字系统不能直接对模拟信号进行处理。但是，许多数字系统又必须输入和处理一些模拟信号，这时就需要经过模-数转换器(A/D)把模拟信号变换成数字系统所能接受的数字信号，同时，数字系统的输出信号又必须经过数-模转换器(D/A)变换成连续的模拟信号。这种 A/D 和 D/A 系统也是数字系统的一部分，其内容在“计算机电路基础”课中已做介绍，本书不再讨论。

3. 课程的任务

本课程的主要任务是讨论数字逻辑电路的分析与设计。研究一个现成的数字逻辑电路的工作原理和逻辑功能叫做“分析”。先确定要完成的逻辑功能，再求出相应的逻辑电路叫做“逻辑设计”或“逻辑综合”。

虽然分析比设计简单，但却非常重要。通过对某些数字逻辑电路的分析，可以汲取其好的设计思想，可以评定其技术性能和经济效益，可以了解不断出现的新数字集成电路的功能，为合理选用数字集成电路、设计最佳数字逻辑电路打好基础。

随着数字集成电路的发展，逻辑设计方法也在发生变化，如用小规模集成电路进行逻辑设计的经典设计方法，在中、大规模集成电路的数字系统设计中已经不适用了，但中、大规模集成电路本身却是通过逻辑设计得到的。因此掌握经典逻辑电路的分析与设计方法，对正确分析和选择使用中、大规模集成电路仍然是重要的。因此，本书以介绍经典的逻辑设计方法为主，同时介绍一些用中规模集成电路进行逻辑设计的特点和方法。为了适应发展的需要，本书介绍了中规模集成逻辑部件的基本功能和扩展应用方法。

由于“数字逻辑”是一门实践性很强的技术基础课，因此在学习本课程时，不仅要重视理论学习，还应注重实践性环节，要完成一定数量的习题和实验，在反复运用中加深理解、巩固所学的知识，只有这样，才能掌握本课程的基本内容，掌握数字逻辑电路的分析方法和设计方法，培养灵活地解决实际问题的能力。

第一章　数制与编码

十进制是我们日常生活和工作中常用的计数制，但在电子计算机和数字系统中广泛使用的却是二进制。本章将介绍十进制与二进制的计数规则及数制转换方法。对于带符号数的代码，主要讨论补码在数字系统中的表示法及加减运算规则。此外介绍几种常见的十进制数的代码表示法及可靠性编码。

1-1　进位计数制

一、十进位计数制

数制是以表示计数值符号的个数（称为基数）来命名的。十进位计数制是人们最熟悉的一种进位计数制。十进位计数制（简称十进制）的主要特点如下。

1. 采用十个数字符号：0、1、2、…、9。这些数字符号称为数码（或数字）。

2. 按"逢十进一"的原则计数。某位数计到 9，再加上 1 等于 10，即 $9+1=10$，本位为"0"，同时向高位进"1"。

3. 数码的位置原理：同一个数码在不同的位置时，所表示的数值不同。例如十进制数为 333，虽然三个数码都是 3，但是最右边的是个位数，因此这一位的数码 3 表示有三个 1，即 3×1；中间的一位是十位数，其数码 3 表示有三个 10，即 3×10；而左边的一位是百位数，其数码 3 表示有三个 100，即 3×100。因此该数读作"三百三十三"，用数学式可表示为

$$\begin{aligned}(333)_{10} &= 3\times100+3\times10+3\times1\\ &= 3\times10^2+3\times10^1+3\times10^0\end{aligned}$$

式中括号外的下标 10 表示十进制（下同）。又例如十进制数 2647 可表示为

$$(2647)_{10}=2\times10^3+6\times10^2+4\times10^1+7\times10^0$$

由此可见，任何一个十进制数都可以写成"以 10 为底的幂的和"这样一种展开形式。10 是十进制的基数，每一位数码对应一个以 10 为底的幂，分别为 10^0、10^1、10^2、10^3 等。每一位 10 的幂称为"位权"，简称"权"。相邻位的权相差十倍，并从右至左递增。

任何一个十进制数 N 按权的展开式可表示为

$$\begin{aligned}(N)_{10} = {} & K_{n-1}\times10^{n-1}+K_{n-2}\times10^{n-2}+\cdots+K_1\times10^1+K_0\times10^0\\ & +K_{-1}\times10^{-1}+\cdots+K_{-m}\times10^{-m}\end{aligned}\qquad(1\text{-}1)$$

或写成和式

$$(N)_{10}=\sum_{i=-m}^{n-1}K_i\cdot(10)^i$$

式中 n 代表整数位数，m 代表小数位数。10^{n-1}、10^{n-2}、…、10^1、10^0、10^{-1}、…、10^{-m}是各位的“位权”，“10”是进位制的基数。K_{n-1}、K_{n-2}、…、K_1、K_0、K_{-1}、…、K_{-m}是十进制数 N 中各位的数码。

例 1　已知$(N)_{10}=3245.16$，将$(N)_{10}$按权展开。

解　该数有六位数码，其中整数为四位数码，$n=4$；小数部分有两位数码，$n=2$。该十进制数的按权展开式写为

$$
\begin{aligned}
(N)_{10} &= K_3 \times 10^3 + K_2 \times 10^2 + K_1 \times 10^1 + K_0 \times 10^0 + K_{-1} \times 10^{-1} + K_{-2} \times 10^{-2} \\
&= 3 \times 10^3 + 2 \times 10^2 + 4 \times 10^1 + 5 \times 10^0 + 1 \times 10^{-1} + 6 \times 10^{-2}
\end{aligned}
$$

又如，将$(N)_{10}=722.5$按权展开后为

$$(N)_{10} = 7 \times 10^2 + 2 \times 10^1 + 2 \times 10^0 + 5 \times 10^{-1}$$

二、二进位计数制及二进制数的四则运算

除十进制数外，在电子计算机中普遍采用基数为 2，数码是 0 和 1 的二进制数。采用二进制数的原因是：它运算简单，容易用数字电路实现，因为这种进位制表示的数，每位仅可能有两种数字符号，特别适合于用电子元件表示。一般说来，制造两个确定状态的器件（这两个状态可以分别表示数码 0 和 1）比制造具有多个确定状态的器件容易得多。因此，在数字电路中，二进制获得极其广泛的应用。

1. 二进位计数制

二进位计数制（简称二进制）的主要特点是：

(1) 采用“0”和“1”两个数码，任何一个二进制数都是由这两个数码表示的（二进制中的“1”一般读作“么”）。

(2) 按“逢二进一”的原则计数，即 1+1=10（读作“么零”）。当一位数计到 1 时，若再加上 1，按十进制应等于 2，但由于二进制中只能用两个数字符号 0、1，所以记作 10。即本位为 0，向高位进 1。这样，十进制数 2 和二进制数的 10 是等值的，记作 $(2)_{10}=(10)_2$。括号外的下标数字表示进位制的基数。二进制数 10 再加 1，记作 11，相当于十进制的 3，记作 $(11)_2=(3)_{10}$。如再加 1，则记作 $(100)_2=(4)_{10}$，依此类推。表 1-1 列出了十进制数与二进制数的对应关系。

因此，在二进制中，同一个数码在数中的位置不同，表示的数值也不相同。由于二进制中只有两个数码，所以进位基数为 2。根据十进制数的按权展开式 (1-1)，可以写出任意二进制数 M 的按权展开式，即

$$
\begin{aligned}
(M)_2 = K_{n-1} \times 2^{n-1} + K_{n-2} \times 2^{n-2} + \cdots + K_1 \times 2^1 \\
+ K_0 \times 2^0 + K_{-1} \times 2^{-1} + \cdots + K_{-m} \times 2^{-m}
\end{aligned}
\tag{1-2}
$$

表 1-1 数制表（整数）

$R=10$	$R=2$	$R=3$	$R=4$	$R=8$	$R=16$
0	0	0	0	0	0
1	1	1	1	1	1
2	10	2	2	2	2
3	11	10	3	3	3
4	100	11	10	4	4
5	101	12	11	5	5
6	110	20	12	6	6
7	111	21	13	7	7
8	1000	22	20	10	8
9	1001	100	21	11	9
10	1010	101	22	12	A
11	1011	102	23	13	B
12	1100	110	30	14	C
13	1101	111	31	15	D
14	1110	112	32	16	E
15	1111	120	33	17	F
16	10000	121	100	20	10

或写成和式

$$(M)_2=\sum_{i=-m}^{n-1}K_i\cdot(2)^i \qquad (1\text{-}3)$$

式中 n 代表整数位数，m 代表小数位数，2^{n-1}、2^{n-2}、…、2^1、2^0、2^{-1}、…、2^{-m}是各位的“位权”，“2”是进位基数，K_{n-1}、K_{n-2}、…、K_1、K_0、K_{-1}…、K_{-m}是二进制数各位的数码。

例 2 将 $(M)_2=1011.101$ 按权展开。

解 $(M)_2=K_3\times2^3+K_2\times2^2+K_1\times2^1+K_0\times2^0+K_{-1}\times2^{-1}+K_{-2}\times2^{-2}+K_{-3}\times2^{-3}$

所以

$$(1011.101)_2=1\times2^3+0\times2^2+1\times2^1+1\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}$$

2. 二进制数的算术运算规则

(1) 加法规则

$$0+0=0$$
$$0+1=1$$
$$1+0=1$$
$$1+1=10$$

在做算术加法运算时，要注意“逢二进一”的原则，即遇到 2 就向相邻的高位进一。

例 3 求 1101+110=?

解

$$\begin{array}{r} 1101 \\ +\ \ 110 \\ \hline 10011 \end{array}$$

所以 $1101+110=10011$

(2) 减法规则

$$0-0=0$$
$$1-0=1$$
$$1-1=0$$

$$10-1=1$$

在做算术减法运算时，若遇到 0 减 1，本位不够，需向高位借 1。在二进制中向高位借 1，在本位作 2 用，即借一作二。

例 4　求 1101－110＝？

解

$$\begin{array}{r} 1101 \\ -\ \ 110 \\ \hline 111 \end{array}$$

所以

$$1101-110=111$$

（3）乘法规则

$$0\times0=0$$
$$0\times1=0$$
$$1\times0=0$$
$$1\times1=1$$

例 5　求 1101×110＝？

解

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

所以

$$1101\times110=1001110$$

（4）除法运算

例 6　求 10010÷110＝？

解

$$\begin{array}{r} 11 \\ 110\,\sqrt{10010} \\ 110\ \, \\ \hline 110 \\ 110 \\ \hline 0 \end{array}$$

所以

$$10010\div110=11$$

二进制数的运算方法与十进制数的运算方法相似。并且由于二进制数中只有两个数码，所以二进制数的运算比十进制数的运算简单。

三、八进制和十六进制

二进制在电子计算机和数字系统中获得了广泛应用，但位数冗长的二进制数读写困难，且容易出错。因此，人们为了简化读写多位二进制数，常采用八进制或十六进制数。

八进制数的基数是 8，采用 0、1、2、3、4、5、6、7 等八个数码。它的 0～7 具有和十进制中的 0～7 一样的数值。

八进制数的计数规律是“逢八进一”，其按权的展开式为

$$(N)_8=\sum_{i=-m}^{n-1}K_i\cdot(8)^i \tag{1-4}$$

由于八进制数的数码和十进制的数码的前八位相同，故约定凡注有下标 8 的数为八进制数，例如 $(123)_8$、$(705)_8$ 等。

十六进制数的基数为 16、采用的数码为：0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。符号 A～F 分别代表十进制数的 10～15。为了区分十进制数和十六进制数，我们约定，凡注有下标 16 的数或最后标有 H 的数为十六进制数。例如 $(123)_{16}$、(123BH)、$(5F7)_{16}$等。

十六进制的计数规律为“逢十六进一”。十六进制数按权的展开式为

$$(N)_{16}=\sum_{i=-m}^{n-1}K_i\cdot(16)^i \tag{1-5}$$

对于任意的 R 进制数 $(N)_R$，也可以用按权的展开式写出

$$\begin{aligned}(N)_R=&K_{n-1}\cdot R^{n-1}+K_{n-2}\cdot R^{n-2}+\cdots+K_1\cdot R^1\\&+K_0\cdot R^0+K_{-1}\cdot R^{-1}+\cdots+K_{-m}\cdot R^{-m}\end{aligned} \tag{1-6}$$

或写成

$$(N)_R=\sum_{i=-m}^{n-1}K_i\cdot(R)^i \tag{1-7}$$

式中 n 代表整数位数，m 代表小数位数，K_i 是 R 进制中 R 个数字符号中的任意一个，即

$$0\leqslant K_i\leqslant(R-1) \tag{1-8}$$

计数时每一位都是“逢 R 进一”。

表 1-1 列出了几种进位制部分整数的表述及它们相互之间的对应关系。

1-2 数制转换

数字系统基本采用二进制，而人们习惯使用十进制，因此经常需要在两者之间进行转换。为方便二进制与十进制之间的转换，在计算机中常采用八进制、十六进制或二十进制等用以间接转换。一个数从一种进位计数制表示转换成另外一种进位计数制表示，称为数制转换。

一、二进制数转换成十进制数

由二进制直接转换成十进制的方法很多，其中最常用的方法有：

1. 按权展开法（加权法）

这种方法是按式(1-3)，把二进制数先写成按权的展开式，然后依照十进制加法规则相加、即得到与二进制数等值的十进制数值。二进制数各位的位权见表 1-2。

例 1 把 $(110101)_2$ 转换成十进制数。

解 $(110101)_2 = 1\times 2^5 + 1\times 2^4 + 0\times 2^3 + 1\times 2^2 + 0\times 2^1 + 1\times 2^0$

$= 32+16+4+1$

$= (53)_{10}$

例 2 把 $(11101101011)_2$ 转换成十进制数。

解 $(11101101011)_2 = 1\times 2^{10} + 1\times 2^9 + 1\times 2^8 + 1\times 2^6 + 1\times 2^5$

$+1\times 2^3 + 1\times 2^1 + 1\times 2^0$

$= 1024+512+256+64+32+8+2+1$

$= (1899)_{10}$

例 3 把 $(0.110101)_2$ 转换成十进制数。

解 $(0.110101)_2 = 1\times 2^{-1} + 1\times 2^{-2} + 0\times 2^{-3} + 1\times 2^{-4} + 0\times 2^{-5} + 1\times 2^{-6}$

$= 0.5+0.25+0.0625+0.015625$

$= (0.828125)_{10}$

表 1-2 二进制各位的权(2^i)

i	2^i	i	2^i	i	2^i
		0	1	7	128
−1	0.5	1	2	8	256
−2	0.25	2	4	9	512
−3	0.125	3	8	10	1024
−4	0.0625	4	16	11	2048
−5	0.03125	5	32	12	4096
−6	0.015625	6	64	13	8192

2. 连乘法

一个二进制数的按权展开式，可以改写成 2 的套乘及套除形式：

$$(N)_2 = \{[(K_{n-1}\times 2 + K_{n-2})\times 2 + \cdots + K_2]\times 2 + K_1\}\times 2 + K_0 + \{[(K_{-m}\times 2^{-1} + K_{-(m-1)})\times 2^{-1} + \cdots + K_{-2}]\times 2^{-1} + K_{-1}\}\times 2^{-1} \quad (1\text{-}9)$$

根据改写后的二进制数的按权展开式(1-9)可知：

若要把二进制数的整数部分转换成为十进制数时，应从最高位算起，将整数部分的最高位数码乘 2，再加上次高位数码，其和再乘 2，再加上下一位数码，这样重复地做下去，直到加上整数部分的最低位数码，即得出变换后的十进制数整数部分。

若要把二进制数的小数部分转换成为十进制数的小数时，是从最低位算起，将小数部分的最低位数码除以 2，随后加上次低位数码，再除以 2，这样重复地做下去，直到加上小数部分的最高位数码后再除以 2，即得变换后的十进制小数部分。

由于二进制数可能包括整数部分和小数部分，变换时可分别将每一部分转换成十进制数，然后将这两部分结合起来，即得到一个完整的十进制数。

例 4 将 $(110101.110101)_2$ 转换成十进制数。

解 转换时，二进制数的整数部分和小数部分分别进行转换。

整数部分为 110101，从最高位开始运算：

$$\begin{array}{rcl}
 & 1 & \text{高位} \\
1\times2+1=3 & 1 & \\
3\times2+0=6 & 0 & \\
6\times2+1=13 & 1 & \\
13\times2+0=26 & 0 & \\
26\times2+1=53 & 1 & \\
 & & \downarrow\ \text{低位}
\end{array}$$

所以
$$(110101)_2=(53)_{10}$$

小数部分为 0.110101，从最低开始运算：

$$\begin{array}{rcl}
1\times\frac{1}{2}=0.5 & 1 & \uparrow\ \text{低位} \\
(0.5+0)\times\frac{1}{2}=0.25 & 0 & \\
(0.25+1)\times\frac{1}{2}=0.625 & 1 & \\
(0.625+0)\times\frac{1}{2}=0.3125 & 0 & \\
(0.3125+1)\times\frac{1}{2}=0.65625 & 1 & \\
(0.65625+1)\times\frac{1}{2}=0.828125 & 1 & \text{高位}
\end{array}$$

所以
$$(0.110101)_2=(0.828125)_{10}$$
$$(110101.110101)_2=(53.828125)_{10}$$

二、十进制数转换成二进制数

十进制数转换成二进制数时，可以应用二进制数转换成十进制数方法的逆运算——“减权法”和“连除法”。

1. 减权法

减权法又叫提取 2 的幂法，步骤是：从十进制数中减去该数中够减的 2 的最高次幂，并在相应权的位置上写 1，如果第一次相减的差数不够减 2 的下一次幂，那就在该相应权的位置上写 0，依此类推，直到十进制数被减为 0 为止。

例 5 把 $(53.5)_{10}$转换成二进制数。

解

$$\begin{array}{rl}
 & 53.5 \\
 & \underline{-32} \longrightarrow 2^5 \\
 & 21.5 \\
 & \underline{-16} \longrightarrow 2^4 \\
 & 5.5 \\
 & \underline{-\ 4} \longrightarrow 2^2 \\
 & 1.5 \\
 & \underline{-\ 1} \longrightarrow 2^0 \\
 & 0.5 \\
 & \underline{-0.5} \longrightarrow 2^{-1} \\
 & 0
\end{array}
\qquad
\begin{array}{ccccccc}
2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 & 2^{-1} \\
1 & 1 & 0 & 1 & 0 & 1 & 1
\end{array}$$

所以
$$(53.5)_{10}=(110101.1)$$

2. 连除法

用此方法转换时，十进制数的整数部分和小数部分也应分别进行转换。

(1) 整数转换

如果将一个十进制数整数部分转换成与其等值的二进制数，则可将该十进制数被 2 除，除法过程还是十进制运算，所得的余数为二进制数的最低位，接着将所得的商再被 2 除，得到的余数为二进制数的次低位。逐次以 2 除上次的商，逐次记下余数，直到其商得 0 为止，这种方法叫“除 2 取余法”。

例 6 把 $(1899)_{10}$转换成二进制数。

解

		余数	
2 | 1899	⟶	1	低位 ↑
2 | 994	⟶	1	
2 | 474	⟶	0	
2 | 237	⟶	1	
2 | 118	⟶	0	
2 | 59	⟶	1	
2 | 29	⟶	1	
2 | 14	⟶	0	
2 | 7	⟶	1	
2 | 3	⟶	1	
2 | 1	⟶	1	
0	⟶		高位

所以 $$(1899)_{10}=(11101101011)_2$$

(2) 小数部分转换

要将十进制数小数部分转换成等值的二进制小数，可用基数 2 连乘小数部分，每次取整数，直到小数部分为 0 或达到所需的位数为止。第一次用基数 2 连乘小数所得的整数部分为 K_{-1} 值，最后一次得到的整数部分为 K_{-m} 值。这种方法又叫“乘 2 取整法”。

例 7 把 $(0.828125)_{10}$转换成二进制数。

解

		整数部分	
$0.828125 \times 2 = 1.656250$	⟶	1	K_{-1} 高位 ↓
$0.656250 \times 2 = 1.312500$	⟶	1	
$0.312500 \times 2 = 0.625000$	⟶	0	
$0.625000 \times 2 = 1.250000$	⟶	1	
$0.250000 \times 2 = 0.500000$	⟶	0	
$0.500000 \times 2 = 1.000000$	⟶	1	K_{-6} 低位

所以 $$(0.828125)_{10}=(0.110101)_2$$