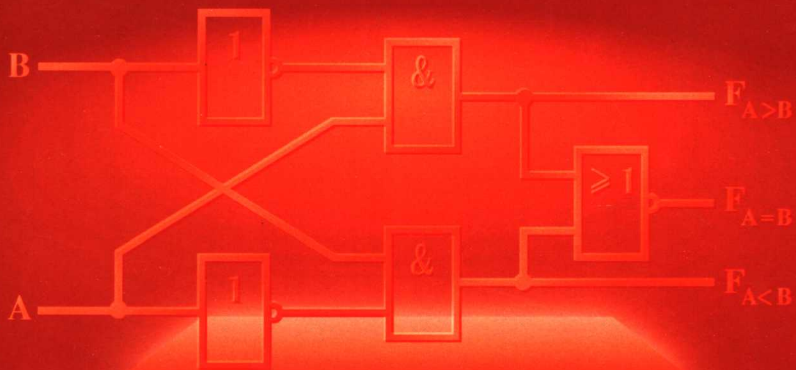
21世纪高等院校电气信息类系列教材



数字电路逻辑设计

李云 许柯 李教 徐有 编著

机械工业出版社
CHINA MACHINE PRESS

21 世纪高等院校电气信息类系列教材

数字电路逻辑设计

李云 许柯 李教 徐有 编著

机械工业出版社

本书全面介绍了数字电路、脉冲电路和数字系统中常用电路及基本模块的工作原理、分析方法及设计方法。全书共10章，包括数制与编码、逻辑函数及其化简、集成逻辑门、组合逻辑电路、集成触发器、时序逻辑电路、半导体存储器、可编程逻辑器件与VHDL基础、脉冲单元电路、数/模与模/数转换器等内容。同时还介绍了一些常用的小规模、中规模、大规模集成器件的功能及应用。每章均选用了较多的典型实例，并配有相当数量的习题。

本书可作为高等院校通信、电子工程、计算机技术、自动控制等专业的技术基础课教材，也可供相关专业的工程技术人员参考。

图书在版编目（CIP）数据

数字电路逻辑设计/李云等编著. —北京：机械工业出版社，2007.6
（21世纪高等院校电气信息类系列教材）
ISBN 978-7-111-21363-5

Ⅰ. 数…　Ⅱ. 李…　Ⅲ. 数字电路—逻辑设计—高等学校—教材　Ⅳ. TN79

中国版本图书馆CIP数据核字（2007）第057205号

机械工业出版社（北京市百万庄大街22号　邮政编码100037）
责任编辑：时　静　版式设计：冉晓华　责任校对：李　婷
责任印制：洪汉军
北京双青印刷厂印刷
2007年7月第1版第1次印刷
184mm×260mm・15.75印张・388千字
0001—5000册
标准书号：ISBN 978-7-111-21363-5
定价：23.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
销售服务热线电话：（010）68326294
购书热线电话：（010）88379639 88379641 88379643
编辑热线电话：（010）88379739

出版说明

随着科学技术的不断进步，整个国家自动化水平和信息化水平的长足发展，社会对电气信息类人才的需求日益迫切、要求也更加严格。在教育部颁布的“普通高等学校本科专业目录”中，电气信息类（Electrical and Information Science and Technology）包括电气工程及其自动化、自动化、电子信息工程、通信工程、计算机科学与技术、电子科学与技术、生物医学工程等专业。这些专业的人才培养对社会需求、经济发展都有着非常重要的意义。

在电气信息类专业及学科迅速发展的同时，也给高等教育工作带来了许多新课题和新任务。在此情况下，只有将新知识、新技术、新领域逐渐融合到教学、实践环节中去，才能培养出优秀的科技人才。为了配合高等院校教学的需要，机械工业出版社组织了这套“21世纪高等院校电气信息类系列教材”。

本套教材是在对电气信息类专业教育情况和教材情况调研与分析的基础上组织编写的，期间，与高等院校相关课程的主讲教师进行了广泛的交流和探讨，旨在构建体系完善、内容全面新颖、适合教学的专业教材。

本套教材涵盖多层面专业课程，定位准确，注重理论与实践、教学与教辅的结合，在语言描述上力求准确、清晰，适合各高等院校电气信息类专业学生使用。

机械工业出版社

前　言

本书是根据国家教委工科电工课程教学指导委员会审订通过的高等院校“电子线路”课程教学的基本要求，并结合我们多年来的教学经验而编写的专业基础课教材。

数字电子技术是当前发展最快的学科之一，随着集成电路工艺的不断发展，数字集成器件已经经历了从小规模集成电路（SSI）、中规模集成电路（MSI）到大规模集成电路（LSI）、超大规模集成电路（VLSI）的发展过程，特别是可编程逻辑器件的出现，为数字电路设计提供了更加完善、方便的器件。相应地，数字电路的设计过程和设计手段也在不断演变和发展，因而对数字电路课程的教学内容、教学方法、教学手段以及教材也提出了新的要求。

本教材以数字集成电路贯穿全篇，突出和加强了数字电路的内容，压缩和精简了脉冲电路部分的内容，在内容的选取上，突出基本概念、基本原理、基本分析方法和工程应用。尽管中、大规模集成电路已成为数字系统的主体，但小规模集成电路仍然是各种类型数字系统中不可缺少的部分，因此，作为数字技术的入门课程，本书以介绍在今后相当长的时期中，仍然行之有效的中、小规模集成电路的分析和设计方法为重点，在确保基本理论、基本概念和基本方法教学的前提下，力求反映当前数字技术的新发展，介绍目前已普遍应用的新器件和已趋于成熟的新技术和新方法。为了加深读者对概念的理解，学以致用，结合作者多年来的教学经验和科研实践体会，书中选择了较多的实例，每章后附有习题，以巩固所学知识，并介绍了一些工程实践中的常用的分析和使用方法，以帮助读者提高解决问题的能力。

通过本书的学习，读者可掌握数字电路及基本脉冲电路的原理和分析、设计方法。能对常见的集成电路进行分析，设计和应用，并能初步掌握可编程逻辑器件的电路结构特点、基本工作原理和开发过程。

本书写作分工如下：徐有编写第1章，李教编写第2章，许柯编写第3、4、9章，李云编写第5、6、7、8、10章。全书由李云修改和统稿。张敏教授审阅全稿，并提出了许多修改意见，在此表示衷心的感谢。

由于编者水平有限，书中错误和不妥之处，请读者批评指正。

编　者

目　录

第 1 章　数制与码制

本章首先介绍数字信号与数字电路的概念，然后从我们熟悉的十进制入手，分析推导出各种不同数制的表示方法以及各种数制之间的转换方法，重点讨论数字计算机以及数字设备中广泛采用的二进制数，最后介绍几种常用的 BCD 码。

1.1　数字信号与数字电路

1. 数字信号和模拟信号

在自然界中，存在着两类物理量：一类是在时间上和数值上都是连续变化的物理量，称为模拟量（Analog Quantity），例如时间、温度、压力等就是模拟量；另一类是在时间上和数值上都是离散的、不连续变化的物理量，称为数字量（Digital Quantity），例如训练场上运动员的人数、车间仓库里元器件的个数等就是数字量。

在电子设备中，无论是数字量还是模拟量都是以电信号形式出现的。人们常常将表示模拟量的电信号叫作模拟信号（Analog Signal），将表示数字量的电信号叫作数字信号（Digital Signal）。正弦波信号、语音信号、图像信号就是模拟信号；矩形波、方波信号就是数字信号。

数字信号是一种脉冲信号（Pulse Signal）。脉冲信号是指一种持续时间极短的电压或电流波，具有边沿陡峭的特点。广义的讲，通常把一切非正弦信号都称为脉冲信号。

2. 数字信号的表示方法

在数字电路中采用只有 0、1 两种数值组成的数字信号。一个 0 或一个 1 通常称为 1 比特，有时也将一个 0 或一个 1 的持续时间称为一拍。数字信号有电位型和脉冲型两种表示方法，电位型数字信号是用电位的高低来表示“1”或“0”，即在一个时间节拍内信号高电平和低电平用“1”和“0”来表示；脉冲型数字信号是用有无脉冲来表示“1”或“0”，即在一个时间节拍内有脉冲和无脉冲用“1”和“0”来表示，如图 1-1-1 所示为数字信号 010011010，图 1-1-1a 为用高电平表示 1，用低电平表示 0 的电位型数字信号；图 1-1-1b 为有脉冲表示 1，无脉冲表示 0 的脉冲型数字信号。

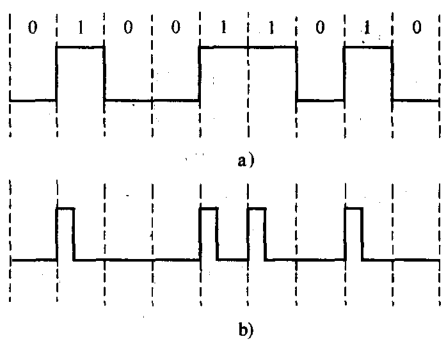


图 1-1-1　数字信号的传输波形
a）电位型信号　b）脉冲型信号

3. 数字电路的特点

在电子电路中，人们将产生和处理模拟信号的电路称为模拟电路（Analog Circuit），将

产生和处理数字信号的电路称为数字电路（Digital Circuit）。

与模拟电路相比，数字电路主要具有以下特点：

1）数字信号通常只有“0”和“1”两种不同的状态，在数字电路中，半导体管多数工作在开关状态，即工作在饱和区和截止区。数字电路的基本单元比较简单，对元器件的精度要求不高，只要能区分出“0”和“1”两种状态就可以了，所以数字电路结构简单，制造容易，便于集成和系列化生产。

2）数字电路不仅可以对信号进行算术运算，而且还能进行逻辑推演和判断，具有一定的逻辑思维能力，这就使它能在数字计算机、数字控制、数据采集和处理、数字通信等领域中获得广泛的应用。

3）因为数字电路研究的主要对象是电路的输入与和输出之间的逻辑关系，所以数字电路也称数字逻辑电路或逻辑电路。其分析方法与模拟电路不同，采用的是逻辑代数、真值表、卡诺图、特性方程、状态转换图、时序波形图等。

4）由数字电路组成的数字系统，具有传输可靠、易于存储、抗干扰能力强、稳定性好等优点，便于使用、维护和进行故障诊断。因此，数字电路获得了越来越广泛的应用。

1.2 数制

1.2.1 进位计数制

数制（Number System）是人类表示数值大小的各种方法的统称。迄今为止，人类都是按照进位方式来实现计数的，这种计数制度称为进位计数制。常用的进位计数制有十进制、二进制、八进制和十六进制。一种数制中允许使用的数符个数称为这种数制的基数（Radix），R 进制的基数就等于 R。例如十进制，每个数位规定使用的数码符号为 0、1、2、…、9 共 10 个，故其基数 $R=10$。

1. 十进制（Decimal）

十进制采用 10 个不同的数码 0、1、2、…、9；进位规则是“逢十进一，借一当十”。

若干个数码并列在一起可以表示一个十进制数。例如数 555.5，小数点左边第一位的 5 代表个位，它的数值为 5；小数点左边第二位的 5 代表十位，它的数值为 5×10^1；左边第三位的 5 代表百位，它的数值为 5×10^2；小数点右边第一位的值为 5×10^{-1}。可见，数码处于不同的位置，代表的数值是不同的。这里 10^2、10^1、10^0、10^{-1} 称为权值或位权值，即十进制数中各位的权值是基数 10 的幂，各位数码的值等于该数码与其权值的乘积。因此对于任何一个十进制数 N，都可以按位权值展开为

$$
\begin{aligned}
(N)_{10} &= a_{n-1}a_{n-2}\cdots a_1a_0\cdots a_{-1}a_{-2}\cdots a_{-m}\\
&= a_{n-1}\times10^{n-1}+a_{n-2}\times10^{n-2}+\cdots+a_1\times10^1+a_0\times10^0+a_{-1}\times10^{-1}+\\
&\quad a_{-2}\times10^{-2}+\cdots+a_{-m}\times10^{-m}\\
&= \sum_{i=-m}^{n-1} a_i\times10^i \qquad (1\text{-}2\text{-}1)
\end{aligned}
$$

式中，n 代表整数位数，m 代表小数位数，$a_i(-m\leqslant i\leqslant n-1)$ 表示第 i 位数码，它可以是 0、1、2、3、…、9 中的任意一个，10^i 为第 i 位数码的权值。

上述十进制数按位权展开的表示方法，可以推广到任意进制数。对于一个基数为 R($R \geqslant 2$)的 R 进制计数制，共有 0、1、…、$R-1$ 个不同的数码，则一个 R 进制的数可以按位权展开为

$$\begin{aligned}(N)_R &= a_{n-1}a_{n-2}\cdots a_1a_0\cdots a_{-1}a_{-2}\cdots a_{-m} \\ &= a_{n-1}\times R^{n-1}+a_{n-2}\times R^{n-2}+\cdots+a_1\times R^1+a_0\times R^0+a_{-1}\times R^{-1}+ \\ &\quad a_{-2}\times R^{-2}+\cdots+a_{-m}\times R^{-m} \\ &= \sum_{i=-m}^{n-1} a_i\times R^i \end{aligned} \tag{1-2-2}$$

2. 二进制（Binary）

二进制采用 2 个不同的数码 0、1；进位规则是“逢二进一，借一当二”。

二进制的基数 $R=2$，每位的权值是 2 的幂次方。任何一个二进制数，根据式（1-2-2）可按位权展开为

$$\begin{aligned}(N)_R &= a_{n-1}a_{n-2}\cdots a_1a_0\cdots a_{-1}a_{-2}\cdots a_{-m} \\ &= a_{n-1}\times 2^{n-1}+a_{n-2}\times 2^{n-2}+\cdots+a_1\times 2^1+a_0\times 2^0+a_{-1}\times 2^{-1}+ \\ &\quad a_{-2}\times 2^{-2}+\cdots+a_{-m}\times 2^{-m} \\ &= \sum_{i=-m}^{n-1} a_i\times 2^i \end{aligned} \tag{1-2-3}$$

例如：

$$\begin{aligned}(1011.011)_2 &= 1\times 2^3+0\times 2^2+1\times 2^1+1\times 2^0+0\times 2^{-1}+1\times 2^{-2}+1\times 2^{-3} \\ &= (11.375)_{10}\end{aligned}$$

3. 八进制（Octal）

八进制采用 8 个不同的数码 0、1、2、3、4、5、6、7；进位规则是“逢八进一，借一当八”。

八进制的基数 $R=8$，每位的权值是 8 的幂次方。任何一个八进制数，根据式（1-2-2）可按位权展开为

$$(N)_8 = \sum_{i=-m}^{n-1} a_i\times 8^i \tag{1-2-4}$$

4. 十六进制（Hexadecimal）

十六进制采用 16 个不同的数码 0~9、A、B、C、D、E、F，其中 A、B、C、D、E、F 6 个数符依次表示 10~15；进位规则是“逢十六进一，借一当十六”。

十六进制的基数 $R=16$，每位的权值是 16 的幂次方。任何一个十六进制数，根据式（1-2-2）可按位权展开为

$$(N)_{16} = \sum_{i=-m}^{n-1} a_i\times 16^i \tag{1-2-5}$$

以上 4 种计数体制的对照关系如表 1-1 所示。

1.2.2 进位计数制之间的转换

一个数可以表示为不同进制的形式。在日常生活中，人们习惯使用十进制数，而在计算机以及数字设备中则采用二进制数，对于 8 位、16 位、32 位二进制数的书写通常采用十六

进制数，因此经常需要在不同数制间进行转换。

表 1-1　几种数制对照表

十进制	二进制	八进制	十六进制
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	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. 任意 *R* 进制数转换为十进制数

将一个二进制数、八进制数或十六进制数转换成十进制数的方法很简单，只要写出该进制的按位权展开式，然后相加，就可得到等值的十进制数。

【例 1-1】 将二进制数 $(11010.011)_2$ 转换为十进制数。

解： $(11010.011)_2 = 1\times 2^4 + 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0 + 0\times 2^{-1} + 1\times 2^{-2} + 1\times 2^{-3}$

$= 16 + 8 + 0 + 2 + 0 + 0 + 0.25 + 0.125 = (26.375)_{10}$

求二进制数的等值十进制数时，将所有值为 1 的数位的位权值相加即可。

【例 1-2】 将八进制数 $(137.504)_8$ 转换为十进制数。

解： $(137.504)_8 = 1\times 8^2 + 3\times 8^1 + 7\times 8^0 + 5\times 8^{-1} + 0\times 8^{-2} + 4\times 8^{-3}$

$= 64 + 24 + 7 + 0.625 + 0 + 0.0078125$

$= (95.6328125)_{10}$

【例 1-3】 将十六进制数 $(AD5.C)_{16}$转换为十进制数。

解： $(AD5.C)_{16} = 10\times 16^2 + 13\times 16^1 + 5\times 16^0 + 12\times 16^{-1}$

$= 2560 + 208 + 5 + 0.75 = (2773.75)_{10}$

2. 十进制数转换为任意 *R* 进制数

将十进制数转换为任意 *R* 进制数，其转换方法是：将整数部分和小数部分分别进行转换，然后再将它们合并起来。

整数部分：除 *R* 取余数法，先得到的是低位，后得到的是高位。

小数部分：乘 R 取整数法，先得到的是高位，后得到的是低位。

把十进制整数 N 转换成 R 进制数的步骤如下：

1）将 N 除以 R，记下所得的商和余数。

2）将上一步所得的商再除以 R，记下所得商和余数。

3）重复做第 2 步，直到商为 0。

4）将各个余数转换成 R 进制的数码，并按照和运算过程相反的顺序把各个余数排列起来，即为 R 进制的数。

【例 1-4】 把十进制数 $(53)_{10}$ 转换成二进制数。

解：

除数	商		余数	
2	53	商	余数	
2	26		1	最低位
2	13		0	
2	6		1	
2	3		0	
2	1		1	
	0		1	最高位

所以 $(53)_{10}=(110101)_2$

【例 1-5】 把十进制数 $(53)_{10}$ 转换成八进制数。

解：

除数	商		余数
8	53	商	余数
8	6		5
	0		6

所以 $(53)_{10}=(65)_8$

把十进制的纯小数 M 转换成 R 进制数的步骤如下：

1）将 M 乘以 R，记下整数部分。

2）将上一步乘积中的小数部分再乘以 R，记下整数部分。

3）重复做第 2 步，直到小数部分为 0 或者满足精度要求为止。

4）将各步求得的整数转换成 R 进制的数码，并按照和运算过程相同的顺序排列起来，即为所求的 R 进制数。

【例 1-6】 将十进制小数 $(0.375)_{10}$ 转换成二进制数。

解：

运算	整数	
0.375		
× 2	整数	
[0.] 750	0	最高位
× 2		
[1.] 500	1	
× 2		
[1.] 000	1	最低位

所以 $(0.375)_{10}=(0.011)_2$

需要说明的是，当十进制小数不能精确转换为二进制小数时，往往需要有一定的精度要

求，例如要求结果保留几位小数或精度达到百分之几。若要求精度达到10%，则需保留4位小数；若要求精度达到1%，则需保留7位小数；若要求精度达到1‰，则需保留10位小数。需要注意，为了减小转换误差，转换时的小数位数应比要求保留的小数位数多1位，然后根据多出的这1位是0还是1决定取舍，基本原则是0舍1入。例如某二进制数$(0.1001)_2$，保留两位小数时结果为$(0.10)_2$，保留3位小数时结果应为$(0.101)_2$。

如果一个十进制数既有整数部分又有小数部分，只要把它们分别进行转换，然后将结果合并即可。

【例1-7】 将十进制小数$(53.375)_{10}$转换成二进制数。

解： 将整数部分和小数部分分别进行转换，然后再将它们合并起来。

$$(53.375)_{10} = (110101.011)_2$$

3. 二进制数转换为十六进制数、八进制数

从表1-1可见，1位八进制数可以用3位二进制数来表示，1位十六进制数可以用4位二进制数来表示，所以二进制数转换成十六进制数（或八进制数）时，十六进制数（或八进制数）与二进制数之间的转换也很方便。

其转换方法是：以二进制数的小数点为起点，分别向左、向右每四位（或三位）分一组。对于小数部分，最低位一组不足四位（或三位）时，必须在有效位右边补0，使其足位。对于整数部分，最高位一组不足位时，可在有效位的左边补0，也可不补。然后，把每一组二进制数转换成等值的十六进制（或八进制）数，并保持原排序。

【例1-8】 将二进制数$(1110111101.10011)_2$转换为八进制数和十六进制数。

解：

二进制	001	110	111	101.	100	110
八进制	1	6	7	5 .	4	6

所以 $(1110111101.10011)_2 = (1675.46)_8$

二进制	0001	1011	1101.	1001	1100
十六进制	1	B	D .	9	C

所以 $(1110111101.10011)_2 = (1BD.9C)_{16}$

【例1-9】 将$(1101101011.101)_2$转换为十六进制数。

解：

二进制	0011	0110	1011.	1010
十六进制	3	6	B .	A

所以 $(1101101011.101)_2 = (36B.A)_{16}$

4. 十六进制数、八进制数转换为二进制数

十六进制（或八进制）数转换成二进制数时，其方法是：把十六进制（或八进制）数的每一位数码分别转换成等值的四位（或三位）二进制数，并保持原排序即可。整数最高位一组左边的0和小数最低位一组右边的0可以省略。

【例1-10】 将十六进制数$(3AF.C8)_{16}$转换为二进制数。

解： $(3AF.C8)_{16} = (0011\ 1010\ 1111.1100\ 1000)_2$

【例1-11】 将八进制数$(137.5)_8$转换为二进制数。

解： $(137.5)_8 = (001\ 011\ 111.101)_2$

1.3 码制

在数字系统中，任何数据和信息都是用若干位0和1组成的二进制代码来表示的。n位二进制码元可以组成2^n种不同的代码，代表2^n种不同的信息和数据。

1.3.1 二-十进制代码

二-十进制代码（BCD码）是用4位二进制码的10种组合来表示十进制数的0~9，简称BCD码（Binary Coded Decimal）。

由于十进制数共有0、1、2、…、9这10个数符，因此至少需要4位二进制码来表示1位十进制数。4位二进制码共有$2^4=16$种组合，在这16种组合中可任选10种来表示十进制数0~9，多余的6个码组称为禁用码，不允许使用。选择方法不同就构成了各种BCD码。几种常用的BCD码如表1-2所示。

表1-2 几种常用的BCD码表

十进制数	8421码	5421码	2421码	余3码	余3格雷码
0	0000	0000	0000	0011	0000
1	0001	0001	0001	0100	0001
2	0010	0010	0010	0101	0011
3	0011	0011	0011	0110	0010
4	0100	0100	0100	0111	0110
5	0101	1000	1011	1000	0111
6	0110	1001	1100	1001	0101
7	0111	1010	1101	1010	0100
8	1000	1011	1110	1011	1100
9	1001	1100	1111	1100	1000

BCD码分有权码和无权码两大类。

1. 有权BCD码

在有权BCD码中，每一个十进制数符均用一个4位二进制码来表示，这4位二进制码中的每一位均有固定权值。

（1）8421BCD码

8421BCD码是最常用的BCD码，选用4位二进制数的前10个码组0000~1001分别代表十进制的0~9这10个数码，余下的1010~1111这6个代码在8421BCD码中不允许出现。8421BCD码的每一位都有固定的权值，按照从左到右的顺序，四位的权值依次为2^3、2^2、2^1、2^0，即8、4、2、1，故称为8421BCD码。对于有权BCD码，可以根据位权展开求得所代表的十进制数。

（2）5421 BCD码和2421 BCD码

5421 BCD码和2421 BCD码也是有权BCD码，它们从高位到低位的权值分别为5、4、2、1和2、4、2、1。这两种有权BCD码中，有的十进制数码存在两种加权方法，例如，

5421 BCD 码中的数码 5，既可以用 1000 表示，也可以用 0101 表示，2421 BCD 码中的数码 6，既可以用 1100 表示，也可以用 0110 表示。这说明 5421 BCD 码和 2421 BCD 码的编码方案都不是惟一的，表 1-2 只列出了一种编码方案。

2421 BCD 码的 10 个数码中，0 和 9、1 和 8、2 和 7、3 和 6、4 和 5 的代码的对应位恰好一个是 0 时，另一个就是 1。我们称 0 和 9、1 和 8 互为反码。因此 2421 BCD 码具有对 9 互补的特点，它是一种对 9 的自补代码（即只要对某一组代码各位取反就可以得到 9 的补码），在运算电路中使用比较方便。

2. 无权 BCD 码

无权 BCD 码每一位没有固定的位权值，因此不能按位权展开来求它们所代表的十进制数。但是，这些代码都有其特点。

(1) 余 3 BCD 码

余 3 BCD 码是 8421 BCD 码的每个码组加 3 $(0011)_2$ 形成的，其中的 0 和 9、1 和 8、2 和 7、3 和 6、4 和 5 各对码组相加均为二进制数 1111，具有这种特性的代码称为自补代码。对应于同样的十进制数字，这种代码比相应的 8421 码多出 3 $(0011)_2$，所以称余 3 BCD 码。例如十进制数 2 在 8421 BCD 码中是 0010，在余 3 BCD 码中就是 $0010+0011=0101$。余 3 BCD 码的各位无固定的位权值，称为无权码。

(2) 余 3 格雷（Gray）码

余 3 格雷码也称余 3 循环码，属于无权 BCD 码。它的主要特点是任何相邻的两个代码之间仅有一位码元不同，如表 1-2 所示。

由于格雷码的这个特性，使其在传输过程中可以降低其产生错误的概率，并且能提高其运行速度。假如两个相邻的十进制数 5 和 6，其相应的二进制码为 0101 和 0110。在用二进制数作加 1 计数时，如果从 5 变 6，二进制码的最低两位都要改变，但实际上两位改变不可能完全同时发生，可能会瞬间出现过程性的错码，若最低位先置 0，然后次低位再置 1，则中间会出现 0101—0100—0110，即出现暂短的误码 0100，这会造成数字系统的逻辑错误，而且使运算速度降低。若采用余 3 格雷码，由 5 $(0111)_2$ 变成 6 $(0101)_2$，只有一位发生变化，就不会出现上述错码，因而杜绝了出现这种错误的可能，而且运算速度会明显提高。

1.3.2 用 BCD 代码表示十进制数

在 BCD 代码中，4 位二进制代码仅表示 1 位十进制数，对一个多位的十进制数进行编码，需要有与十进制位数相同的几组 BCD 代码来表示，每组代码之间按十进制进位。

8421 BCD 码和十进制数之间的转换是一种直接按位（或组）的转换，即一组 4 位二进制数码代表 1 位十进制数。用 BCD 码表示十进制数时，只要把十进制数的每一位数码分别用 BCD 码取代即可。反之，若要知道 BCD 码代表的十进制数，只要把 BCD 码以小数点为起点向左、向右每四位分一组，再写出每一组代码代表的等值十进制数，并保持原排序即可。

【例 1-12】 将十进制数 $(13.9)_{10}$ 转换成 8421 BCD 码和余 3 BCD 码。

解： 十进制数 0 ~ 9 和 8421 BCD 码的 0000 ~ 1001 一一对应；十进制数 0 ~ 9 每位加 3 按位权值 8421 可得到相应的余 3 BCD 码。

$$(13.9)_{10}=(0001\ 0011.\ 1001)_{8421\ \mathrm{BCD}}=(0100\ 0110.\ 1100)_{\text{余}3\ \mathrm{BCD}}$$

【例 1-13】 $(0001\ 0111\ 0101\ 0000)_{8421\ \mathrm{BCD}}=(\ ?\ )_{10}$

解：$(0001\ 0111\ 0101\ 0000)_{8421\ BCD} = (1750)_{10}$

若把一种 BCD 码转换成另一种 BCD 码，应先求出某种 BCD 码代表的等值十进制数，再将该十进制数转换成另一种 BCD 码。

【例 1-14】 $(0100\ 1000.1011)_{余3\ BCD} = (\ ?\ )_{8421\ BCD}$

解：$(0100\ 1000.1011)_{余3\ BCD} = (15.8)_{10} = (10101.1000)_{8421\ BCD}$

1.4 本章小结

本章所讲的主要内容是数字信号及数字信号的两种表示方法、数制及其相互转换和常用 BCD 代码。在进位计数制中，必须熟练掌握二进制数和十进制数、二进制数和十六进制数之间的相互转换关系。在码制中，必须熟练掌握常用 BCD 码（8421 BCD 码、5421 BCD 码、余 3 BCD 码、余 3 格雷码）和十进制数 0 ~ 9 之间的对应关系，熟练掌握 8421 BCD 码、余 3 BCD 码和余 3 格雷码之间的转换关系。

1.5 习题

1. 把下列二进制数转换成十进制数：

（1）101101　　（2）0.01101　　（3）101001.101

2. 把下列十进制数转换成二进制数：

（1）51　　（2）12.34　　（3）105.375

3. 把下列各位数转换成十进制数：

（1）$(101.1)_2$　　（2）$(3FCA)_{16}$　　（3）$(78.8)_{16}$

4. 完成数制转换：

（1）$(163.27)_{10} = (\ ?\ )_2 = (\ ?\ )_{16}$

（2）$(3AB6)_{16} = (\ ?\ )_{10} = (\ ?\ )_2 = (\ ?\ )_{8421\ BCD}$

5. 完成下列各数的转换：

（1）$(73.26)_{10} = (\ ?\ )_{8421\ BCD} = (\ ?\ )_{余3\ BCD}$

（2）$(1000010110010111)_{8421\ BCD} = (\ ?\ )_{10}$