★高等院校21世纪新视野教材★

陆秀令　陈罗湘　黄建科　主编

数字电子技术

湖南大学出版社

★高等院校21世纪新视野教材★

数字电子技术

主　编　陆秀令　陈罗湘　黄建科

编　委　黄建科　陈罗湘　陆秀令

张榕华　胡新晚　宋绍民

廖代文　范　进　王怡东

曹　斌　伍进革

湖南大学出版社

2004年·长沙

图书在版编目(CIP)数据

数字电子技术/陆秀令,陈罗湘,黄建科主编.—长沙:湖南大学出版社,2004.2

ISBN 7—81053—720—2

Ⅰ.数… Ⅱ.①陆…②陈…③黄… Ⅲ.数字电路—电子技术—高等学校—教材 Ⅳ.TN79

中国版本图书馆 CIP 数据核字(2004)第 002833 号

数字电子技术

Shuzi Dianzi Jishu

陆秀令 陈罗湘 黄建科 主编

□ 责任编辑 厉 亚
□ 特约编辑 何智辉
□ 封面设计 张 毅
□ 出版发行 湖南大学出版社
社址 长沙市岳麓山 邮码 410082
电话 0731—8821691 0731—8821315
□ 经 销 湖南省新华书店
□ 印 装 湖南新华印刷集团有限责任公司(邵阳)

□ 开本 787×1092 16 开 □ 印张 15.75 □ 字数 366 千
□ 版次 2004 年 2 月第 1 版 □ 2004 年 2 月第 1 次印刷
□ 印数 1～5 000 册
□ 书号 ISBN 7—81053—720—2/TN·13
□ 定价 19.00 元

(湖南大学版图书凡有印装差错,请向承印厂调换)

目　　次

第3章 逻辑门电路

第4章　组合逻辑电路

第5章　触发器

第6章　时序逻辑电路

第7章　脉冲波形的产生和整形

第 8 章 存储器和可编程逻辑器件

第 9 章 数/模和模/数转换

第 1 章　数制与编码

1.1　数制

1.1.1　进位计数制

所谓数制就是计数的方法。在生产实践中，人们经常采用不同基数作为计数体制。例如用来计时的有六十进制、二十四进制、三十(或三十一)进制、十二进制等，但最常用的还是十进制，即以 10 为基数的计数体制。一般来说，各种计数体制都有采用位置计数法。即以特定的一些数字符号(也称数码)排列起来，每个符号处于不同位置作为各位的系数，每个位置都有一定的位权。其数值(数字量)就是把各位的位权乘以该位的系数相加之和。

1. 十进制

十进制是以 10 为基数的计数体制，各位的系数为 0、1、2、4、5、6、7、8、9 中的一个，各位的位权是以 10 为底的幂。如 $5350.23=5\times10^3+3\times10^2+5\times10^1+0\times10^0+2\times10^{-1}+3\times10^{-2}$式中是根据每一数码所在的位置而定的，所以称之为位权，十进制中的数码为 0、1、2……9 十个，其进位规律是逢十进一，即 9+1=10。

由此，可以得出十进制数的一般的表达式。如果一个十进制数包含 n 位整数和 m 位小数，则

$$\begin{aligned} N_{10} &= a_{n-1}\times10^{n-1}+a_{n-2}\times10^{n-2}+\cdots+a_1\times10^1+a_0\times10^0+a_{-1}\times10^{-1} \\ &\quad +a_{-2}\times10^{-2}+\cdots+a_{-m}\times10^{-m} \\ &= \sum_{i=-m}^{n-1}a_i\times(10)^i \qquad (1.1.1) \end{aligned}$$

式(1.1.1)中每一项都等于系数 a_i乘以该位所具有的权，即 10 的 i 次幂，整数部分的权为 10 的正幂，小数部分的权为 10 的负幂，乘上了权的系数称为加权系数。这样，十进制的数值等于各加权系数之和。

十进制的位权关系列于表 1.1.1。

表 1.1.1　十进制的位权

位号	$n-1$	$n-2$	…	2	1	0	小数点	-1	-2	…	$-m$
位权	10^{n-1}	10^{n-2}	…	10^2	10^1	10^0	•	10^{-1}	10^{-2}	…	10^{-m}

2. 二进制

在数字电路中常采用的是二进制(Binary)，因为二进制的两个数码 0 和 1 可以与电路的两个状态(饱和与截止)直接对应。二进制是以 2 为基数的计数体制，其计数方法是逢二进一(或借一当二)，其位权为 2 的整数幂，其按权展开式的规律与十进制相同，表

1.1.2列出了二进制各位的位权。

表 1.1.2　各位二进制数的位权

位号	n	0	1	2	3	4	5	6	7	8	9	10	11	12	13
位权	2^n	1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192

例如 $(1011)_2 = (1011)_B$　　（脚号 B 表示二进制数）

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

再如 $(10011.01)_2$

$$=1\times 2^4 + 0\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0 + 0\times 2^{-1} + 1\times 2^{-2}$$

二进制的按权展开式可写作

$$N_2 = \sum_{i=-m}^{n-1} a_i \times 2^i$$

3. 八进制和十六进制

八进制是以 8 为基数的计数体制，用 0、1、2、3、4、5、6、7 这八个数码作系数，各位的位权是 8 的整数幂，其计数规律是逢八进一（或借一当八），其按权展开式为

$$N_8 = N_O = \sum_{i=-m}^{n-1} a_i \times 8^i$$

由于八进制的数码和十进制的前八个数码相同，但其位权基数不同，所以书写时要标注下标 8 或在括号外右下角标注英文字母 O（代表 Octad number）。

如　　$(174)_8 = 1\times 8^2 + 7\times 8^1 + 4\times 8^0 = (124)_{10}$

十六进制是以 16 为基数的计数体制，它采用的数码为 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。符号 A～F 分别代表十进制的 10～15。为了便于区别十进制数和十六进制数，人们规定，凡注有下标 16 或 H 数为十六进制数（H 代表 Hexadecimal number）。其按权展开式为

$$N_{16} = N_H = \sum_{i=-m}^{n-1} a_i \times (16)^i$$

如　　$(4E6)_{16} = (4E6)_H = 4\times 16^2 + E\times 16^1 + 6\times 16^0$

$$= 4\times 16^2 + 14\times 16^1 + 6\times 16^0 = (1254)_{10}$$

上述几种数制各有其优缺点，应用场合也不相同。以十进制和二进制作比较，十进制在日常生活中应用最多，是人们最熟悉和习惯的计数体制，但其十个数码在数字电路中难于找到十个状态与之对应。数字电路的两个状态可用两个数码表示，故采用二进制。二进制计算规则简单，但人们对它不习惯，另外其数位较多，不易读写。八进制和十六进制可借助与二进制的关系，根据需要对十进制数进行不同的编码，也是很有用的。

1.1.2　几种进位计数制之间的转换

1. 非十进制转换为十进制

可以将非十进制数写为按权展开式，得出其相加的结果，就是与其对应的十进制数。

［例 1.1.1］　$(11010)_2 = 1\times 2^4 + 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0$

$$=2^4+2^3+2^1=(26)_{10}$$

[例 1.1.2] $(1001.01)_2=1\times2^3+0\times2^2+0\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2}$

$$=2^3+2^0+2^{-2}=(9.25)_{10}$$

[例 1.1.3] $(174)_8=1\times8^2+7\times8^1+4\times8^0$

$$=64+56+4=(124)_{10}$$

[例 1.1.4] $(4E6)_{16}=4\times16^2+14\times16^1+6\times16^0$

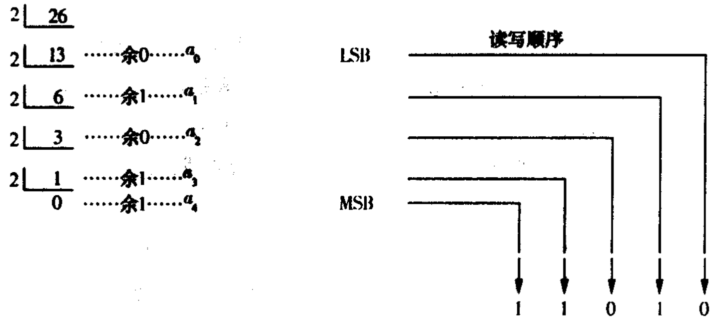
$$=4\times256+224+6=(1254)_{10}$$

2. 十进制转换为非十进制

(1)整数部分的转换

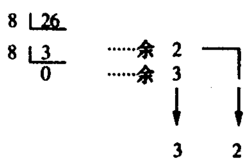
整数部分可以采用连除法，即将原十进制数连续除以转换计数体制的基数，每次除完所得余数就作为要转换数的系数。先得到的余数为转换数的低位，后得到的为高位，直到除得的商为 0 为止。这种方法概括起来可说成"除基数、得余数、作系数，从低位到高位"。符号 LSB 表示最低位，符号 MSB 表示最高位。

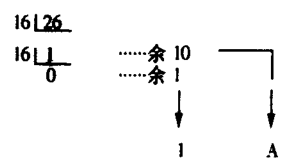
[例 1.1.5] $(26)_{10}=(\quad)_2=(\quad)_8=(\quad)_{16}$



所以$(26)_{10}=(11010)_2$

同理





所以$(26)_{10}=(32)_8=(1A)_{16}$

[例 1.1.6] 将$(81)_{10}$转换为二进制、八进制、十六进制。

先将$(81)_{10}$利用连除法转化为二进制数。

2 | 81	1	LSB
2 | 40	0	
2 | 20	0	
2 | 10	0	
2 | 5	1	
2 | 2	0	
2 | 1	1	MSB
0		

得　　$(81)_{10}=(1010001)_2$

可用连除法求八进制数、十六进制数。也可以利用八进制数码与二进制数码的对应关系，由二进制数直接转化为八进制数。因为每一个八进制数码都可以用 3 位二进制数表示，所以可将二进制数从低位向高位每 3 位一组写出各组的值，从左到右读写，就是八进制数。在将二进制数按 3 位一组划分字节时最高位一组数不够可用 0 补齐。

$$(81)_{10}=(1010001)_2=(001\ 010\ 001)_2$$
$$=(121)_8$$

同理，每一个十六进制数码都可以用 4 位二进制数来表示，所以可将二进制数从低位开始每 4 位一组写出其值，从左到右读写，就是十六进制数。

$$(81)_{10}=(1010001)_2=(0101\ 0001)_2=(51)_{16}$$

小数点以后的二进制数转化为八进制、十六进制数在划分字节时是从高位到低位进行的。

上述方法是可逆的，将八进制的每一位写成 3 位二进制数，左右顺序不变，就能从八进制直接转换为二进制；将十六进制的每一位定成 4 位二进制数，就能从十六进制直接转化为二进制；借助于二进制也可以实现八进制和十六进制的相互转换。

[例 1.1.7]　$(257)_8=(010\ 101\ 111)_2$
$$=(1010\ 1111)_2$$
$$=(AF)_{16}$$

表 1.1.3 给出了十六进制数码与其他几种数制的对照表。

表 1.1.3　数制对照表

计算体制基数 R	数码表示方法
$R=16$	0 1 2 3 4 5 6 7 8 9 A B C D E F 10
$R=2$	0 1 10 11 100 101 110 111 1000 1001 1010 1011 1100 1101 1110 1111 10000
$R=10$	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
$R=8$	0 1 2 3 4 5 6 7 10 11 12 13 14 15 16 17 20

（2）小数部分的转换

十进制小数部分转换为其他进制小数可采用连乘法，即将原十进制纯小数乘以要转换出的基数，取其积的整数部分作系数，剩余的纯小数部分再做基数，先得到的整数作新数的高位（MSB），后得到的作低位，直至其纯小数部分为 0 或到一定精度为止。这种方法可概括地说成“乘基数，取整数，作系数，从高位，到低位”。

[例 1.1.8] 将$(0.875)_{10}$转换为二进制数。

$0.875\times 2 = 1.750 \cdots\cdots 1$ MSB

$0.750\times 2 = 1.500 \cdots\cdots 1$

$0.500\times 2 = 1.000 \cdots\cdots 1$ LSB

所以 $(0.875)_{10}=(0.111)_2$

[例 1.1.9] $(0.78125)_{10}=(\quad)_2$

$0.78125\times 2 = 1.5625 \cdots\cdots 1$ MSB

$0.5625\times 2 = 1.1250 \cdots\cdots 1$

$0.1250\times 2 = 0.25 \cdots\cdots 0$

$0.25\times 2 = 0.5 \cdots\cdots 0$

$0.5\times 2 = 1.0 \cdots\cdots 1$ LSB

所以 $(0.78125)_{10}=(0.11001)_2$

如要求转换为八进制和十六进制,可利用八进制和十六进制与二进制的关系将二进制划分字节获得。对本例有:

$$
\begin{aligned}
(0.78125)_{10} &= (0.11001)_2 \\
&= (0.110\ 010) = (0.62)_8 \\
&= (0.1100\ 1000) = (0.C8)_{16}
\end{aligned}
$$

3. 2^n进位制数之间的转换

(1) 二进制数与八进制数之间的转换

因为3位二进制数能表示十进制数的0~7八个数,这八个数正好符合八进制的8个数码,所以3位二进制数可以表示1位八进制数,因此二进制和八进制数之间的转换是十分简便的。

整数部分:将二进制整数部分自右至左每3位分为一组,最后不足3位时左边用0补足。再把每3位二进制数对应的八进制数码写出即可。

小数部分:将二进制小数部分自左至右每3位分为一组,最后不足3位时在右面用0补足。再把每3位二进制数对应的八进制数码写出即可。

[例 1.1.10] 将$(1011101.01)_2$转换成八进制数。

$$
\frac{001}{1}\quad\frac{011}{3}\quad\frac{101}{5}\cdot\quad\frac{010}{2}
$$

$(1011101.01)_2=(135.2)_8$

若将一个八进制数转换成二进制数,只要写出每位数码所对应的二进制数,依次排好即可。

[例 1.1.11] 将$(364.474)_8$转换为二进制数。

$$
\begin{array}{ccccccc}
3 & 6 & 4 & \cdot & 4 & 7 & 4 \\
\downarrow & \downarrow & \downarrow & \cdot & \downarrow & \downarrow & \downarrow \\
011 & 110 & 100 & & 100 & 111 & 100
\end{array}
$$

$(364.474)_8=(11110100.1001111)_2$

(2) 二进制数与十六进制数之间的转换

同前面二、八进制数的转换原理一样，这里每 4 位二进制数相当 1 位十六进制数，因此当二进制数转换成十六进制数时，只要将二进制数的整数部分自右向左每 4 位分为一组，最后不足 4 位时用 0 补足；小数部分则自左至右每 4 位分为一组，最后不足 4 位时用 0 补足。如若将十六进制数转换成二进制数，则用相反方法进行即可。

[例 1.1.12] 将$(1011101.01)_2$转换成十六进制数。

$$\underset{5}{\underline{0101}}\quad \underset{D}{\underline{1101}}\quad \underset{.}{\underline{.}}\underset{4}{\underline{0100}}$$

$(1011101.01)_2=(5D\cdot 4)_{16}$

[例 1.1.13] 将$(B2A\cdot 5C)_{16}$转换成二进制数。

B　2　A　.　5　C

1011　0010　1010　.　0101　1100

$(B2A.5C)_{16}=(101100101010.010111)_2$

若要将八进制和十六进制数之间进行转换，均可通过二进制数作为转换媒介。

1.2 编码

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 种不同的信息或数据。因此，用若干位二进制码元按一定规律排列起来表示给定信息的过程称为编码。下面介绍数字系统中常用的编码及特性

1.2.1 二-十进制编码(BCD 码)

二-十进制编码是用四位二进制码的 10 种组合表示十进制数 0～9，简称 BCD 码(Binary Coded Decimal)。

这种编码至少需要用四位二进制码元，而四位二进制码元可以有 16 种组合。当用这些组合表示十进制数 0～9 时，有六种组合不用。由 16 种组合中选用 10 种组合，有

$$A_{16}^{10}=\frac{16!}{(16-10)!}\approx 2.9\times 10^{10}$$

种编码方案，但并不是所有的方案都有实用价值，表 1.2.1 列出了几种常用 BCD 码的编码方式。

表 1.2.1　几种常用的 BCD 码

十进制数	8421 码	5421 码	2421 码	余 3 码	BCD Gray 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	01110	1001	1100	1001	0101
7	0111	1010	1101	1010	0100
8	1000	1011	1110	1011	1100
9	1001	1100	1111	1100	1000

1. 8421 BCD码

8421 BCD码是最基本和最常用的BCD码，它和四位自然二进制相似，各位的权值为8、4、2、1，故称为有权BCD码。和四位自然二进制码不同的是，它只选用了四位二进制码中前10组代码，即用0000～1001分别代表它所对应的十进制数，余下的6组代码不用。

2. 5421 BCD码和2421 BCD码

5421 BCD码和2421 BCD码为有权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1.2.1只列出了一种编码方案。

表1.2.1中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、2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。

3. 余3码

余3码是8421 BCD码的每个码组加3(0011)形成的。余3码也具有对9互补的特点，即它也是一种9的自补码，所以也常用于BCD码的运算电路中。

用BCD码可以方便地表示多位十进制数，例如十进制数$(579.8)_{10}$可以分别用8421 BCD码、余3码表示为

$$
\begin{aligned}
(579.8)_{10} &= (0101\ 0111\ 1001.1000)_{8421\,\text{BCD码}} \\
&= (1000\ 1010\ 1100.1011)_{\text{余3码}}
\end{aligned}
$$

1.2.2 可靠性编码

代码在形成、传输过程中可能会发生错误。为了减少这种错误，出现了可靠性编码，其中常用的有两种。

1. Gray码(格雷码)

Gray码也称循环码，其最基本的特性是任何相邻的两组代码中，仅有一位数码不同，因而又叫单位距离码。

Gray码的编码方案有多种，典型的Gray码如表1.2.2所示。从表1.2.2中看出，这种代码除了具有单位距离的特点外，还有一个特点就是具有反射特性，即按表中所示的对称轴为界，除最高位互补反射外，其余低位数沿对称轴镜像对称。利用这一反射性可以方便地构成位数不同的Gray码。

Gray码的单位距离特性有很重要的意义。假如两个相邻的十进制数13和14，相应的二进制码为1101和1110。在用二进制数作加1计数时，如果从13变到14，二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，若最低位先置0，然后次低位再置1，则中间会出现1101—1100—1110，即出现短暂的误码1100，而Gray码因只有一位变化，因而杜绝了出现这种错误的可能。

Gray码是一种具有单位距离特性的BCD码，其编码方案也很多，表1.2.2最右边仅

列出了一种，它有前九组代码与典型的四位 Gray 码相同，仅最后一组代码不同，用 1000 代替了 Gray 码的 1101，这是因为从最大数 9 返回到 0，也应具有单位距离选特性。

2. 奇偶校验码

代码（或数据）在传输和处理过程中，有时会出现代码中的某一位由 0 错变为 1，或 1 变为 0。奇偶校验码是一种具有检验出这种错误的代码，奇偶校验码由信息位和一位奇偶检验位两部分组成。

表 1.2.2 典型的 Gray 码

十进制数	二进制码	Gray 码	
	B_3 B_2 B_1 B_0	G_3 G_2 G_1 G_0	
0	0 0 0 0	0 0 0 0	… 一位反射对称轴
1	0 0 0 1	0 0 0 1	… 二位反射对称轴
2	0 0 1 0	0 0 1 1	
3	0 0 1 1	0 0 1 0	… 三位反射对称轴
4	0 1 0 0	0 1 1 0	
5	0 1 0 1	0 1 1 1	
6	0 1 1 0	0 1 0 1	
7	0 1 1 1	0 1 0 0	… 四位反射对称轴
8	1 0 0 0	1 1 0 0	
9	1 0 0 1	1 1 0 1	
10	1 0 1 0	1 1 1 1	
11	1 0 1 1	1 1 1 0	
12	1 1 0 0	1 0 1 0	
13	1 1 0 1	1 0 1 1	
14	1 1 1 0	1 0 0 1	
15	1 1 1 1	1 0 0 0	

信息位是位数不限的任一种二进制代码。

检验位仅有一位，它可以放在信息位的前面，也可以放在信息位的后面。它的编码方式有两种：

使得一组代码中信息位和检验位中“1”的个数之和为奇数，称为奇检验；

使得一组代码中信息位和检验位中“1”的个数之和为偶数，称为偶检验。

表 1.2.3 给出了 8421 BCD 码的奇偶校验码。

表 1.2.3 带奇偶检验的 8421 BCD 码

十进制数	8421 BCD 奇检验	8421 BCD 奇检验
0	0000 1	0000 0
1	0001 0	0001 1
2	0010 0	0010 1
3	0011 1	0011 0
4	0100 0	0100 1
5	0101 1	0101 0
6	0110 1	0110 0
7	0111 0	0111 1
8	1000 0	1000 1
9	1001 1	1001 0
	信息位 检验位	信息位 检验位