邮电高等函授教材

数字电路与逻辑设计

SHUZI DIANLU YU LUOJI SHEJI

赵六骏
金良玉 编

北京邮电大学出版社

数字电路与逻辑设计

赵六骏　金良玉　编

北京邮电大学出版社

(京)新登字 162 号

内容提要

本书是以邮电高等函授《数字电路与逻辑设计教学大纲》为依据，结合教学实践，以及近几年国内外数字技术的发展情况编写而成的。

全书分为八章，分别介绍了数字逻辑基础、逻辑门电路、组合逻辑电路、集成触发器、时序逻辑电路、大规模集成电路、脉冲信号的产生与波形变换以及数模和模数转换。

本书深入浅出地阐述了数字、逻辑电路的工作原理和分析、设计方法。重点讲述了目前数字系统中常用的中、大规模集成部件的工作原理和应用。每章中有学习要求，有较多的例题、思考题和习题，章末有学习小结，以帮助巩固所学内容。

本书是邮电高等函授(本科)的专用教材，也可作为工科院校通信类、计算机类、无线电类，自动化类等专业数字电子技术课程的教材或教学参考书，还可作为从事数字电子技术的工程技术人员的参考用书。

图书在版编目（CIP）数据

数字电路与逻辑设计/赵六骏，金良玉编. 一北京：北京邮电大学出版社，1995

ISBN 7-5635-0207-6

Ⅰ. 数… Ⅱ. ①赵… ②金… Ⅲ. ①数字电路-数字集成电路-逻辑设计 ②逻辑电路-逻辑集成电路-逻辑设计 Ⅳ. ①TN431. 202 ②TN79

中国版本图书馆 CIP 数据核字（94）第 00827 号

数字电路与逻辑设计

作　　者　赵六骏　金良玉

责任编辑　时友芬

*

北京邮电大学出版社出版

新华书店北京发行所发行　各地新华书店经售

河北省高碑店市印刷厂印刷

*

787×1092 毫米　1/16　印张 18.5　字数 456 千字

1995 年 2 月第一版　　1995 年 2 月第一次印刷

印数：1-8500 册

ISBN 7-5635-0207-6/TN・79　　定价：15.30 元

前　言

本书是以电子技术课程教学指导小组于1992年提出的《高等工业学校电子技术基础课程教学基本要求》为指导，按照邮电高等函授《数字电路与逻辑设计》教学大纲编写而成的。

书稿经专家审评，由邮电高等函授教学指导委员会推荐出版。主要作为邮电高等院校电类专业的函授自学教材。

编者结合多年从事该课程教学的经验，在编写过程中尽量做到文字叙述通俗易懂、逻辑性强；内容安排由浅入深、循序渐近；理论联系实际、多举实例；习题与课程内容相匹配，以便于自学。同时考虑到当前数字电子技术飞速发展的客观现实，在内容取舍上按以下原则进行：

1. 在保证基本理论完整性的原则下，压缩了集成逻辑门、集成触发器和集成的脉冲单元电路中内部电路的分析，突出了对它们的外部特性分析和实际应用。

2. 全书以数字集成电路为主，突出了中、大规模集成电路的分析与应用。书中的主要章节（组合及时序逻辑电路）都以中规模集成电路贯穿全章，并重点阐述了应用中规模集成电路实现逻辑功能的方法。大规模集成电路则单列一章，以便系统讨论。并专设一节，重点介绍新型的可编程逻辑器件PAL和GAL的基本知识。最后一章也以介绍大规模D/A及A/D转换片为主。适当删减了以集成门为基础进行逻辑设计的内容。

3. 加强了CMOS集成电路的内容。

4. 鉴于国家标准局要求自1990年元月1日起，所有电气技术文件和图纸一律使用新的国家标准，而其中与本书关系最大的《二进制逻辑单元》（GB4728. 12-85）标准和旧标准差别甚大，不少教师和技术人员可能还不大习惯，所以本书在采用新逻辑符号的同时，也对比画出了过去常用的旧符号。

5. 本书在每章的开头都编写了学习要求，指明这一章的重点、难点和应掌握的内容。在每章后编写了小结和习题，以帮助学生巩固所学的知识。

本书第一、二、三、七章由全良玉编写，第四、五、六、八章由赵六骏编写。

由于编者的水平所限，缺点和错误在所难免，恳请读者批评指正。

编　者　1995年2月

目　　录

第三章　组合逻辑电路

第四章　集成触发器

第五章　时序逻辑电路

第一章　数字逻辑基础

学 习 要 求

本章主要讲述数字系统中分析和设计逻辑电路的基础知识以及化简逻辑函数的基本方法。

学完本章后，要求：

(1) 理解“位权”的概念，从而能写出任意进制数的按权展开式。掌握二进制数的算术运算规则以及几种常用数制之间的转换方法。

(2) 熟悉 8421、2421，余 3BCD 码和循环码的编码方法以及二进制数与循环码之间的转换方法。

(3) 掌握逻辑代数的基本定律、规则和逻辑函数的描述方法以及异或函数的运算规则。

(4) 掌握逻辑函数表达式与真值表之间的转换、逻辑函数五种一般形式之间的转换和求逻辑函数最小项表达式的方法。

(5) 掌握求逻辑函数的对偶式、反函数以及利用代数法求函数的最简与或式的方法。

(6) 理解最小项、相邻项的概念，掌握四变量以内函数的卡诺图化简法。

1.1　数　　制

在日常生活中，人们习惯使用十进制记数法，而在计算机和数字通信设备中却广泛使用二进制记数法，为了便以人和机器的联系，就必须解决不同数制之间的转换技术。

1.1.1　二－十进制数

1. 十进制数

十进制数的特点是：

(1) 每一位上可出现 0，1，2，3，4，5，6，7，8，9，十进制数码中的任何一个数。例如十进制数 239.85。

(2) 是逢十进一的数，即它的基数为十(即 10)。在不同数位上的数码其值是不同的。例如十进制数 239.85 可改写成：

$$(239.85)_{10}=2\times10^2+3\times10^1+9\times10^0+8\times10^{-1}+5\times10^{-2}$$

上式各数位上的乘数（即 10^2，10^1，10^0，10^{-1}，10^{-2}）称为各相应位的“位权”（简称为“权”），与位权相乘之数称为系数。因此，任意一个十进制数的按“权”展开式为

$$\begin{aligned}(S)_{10} &= K_{n-1}(10)^{n-1} + K_{n-2}(10)^{n-2} + \cdots + K_0(10)^0 + K_{-1}(10)^{-1} + \cdots + K_{-m}(10)^{-m} \\ &= \sum_{i=-m}^{n-1} K_i(10)^i \qquad (1\text{-}1)\end{aligned}$$

式中 K_i 为相应位的数码，n 为小数点前的位数，m 为小数点后的位数。由此可知，十进制就是以 10 为基数的计数体制。

2. 二进制数

在数字通信和计算机系统中广泛采用二进制数，二进制数的特点是：

(1) 每一位上出现的数码仅有 0 和 1 两个数码。

(2) 是逢二进一的数，即它的计数基数为 2。所以二进制数各位的“权”是基数 2 的幂。一个任意的二进制数 $(S)_2$ 它的按权展开式为：

$$(S)_2 = K_{n-1}(2)^{n-1} + K_{n-2}(2)^{n-2} + \cdots + K_0(2)^0 + K_{-1}(2)^{-1} + \cdots + K_{-m}(2)^{-m}$$
$$= \sum_{i=-m}^{n-1} K_i(2)^i \qquad (1\text{-}2)$$

式中符号 K_i、n、m 的定义与十进制数中的定义完全相同。

通过上述两种数制的分析可知，二者的按“权”展开式，除基数不同外，形式上是完全相似的。经分析，对于一个任意进制数 $(S)_J$，它的基数就是 J(J 为任意正整数)，所以它的按权展开式为

$$(S)_J = \sum_{i=-m}^{n-1} K_i(J)^i \qquad (1\text{-}3)$$

例如：八进制数的按权展开式为：

$$(S)_8 = \sum_{i=-m}^{n-1} K_i(8)^i \qquad (1\text{-}4)$$

八进制数中采用的数码是 0，1，2，3，4，5，6，7。

十六进制数的按权展开式为：

$$(S)_{16} = \sum_{i=-m}^{n-1} K_i(16)^i \qquad (1\text{-}5)$$

十六进制数中采用的数码是 0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F。数码 A～F 分别表示十进制数 10～15。

1.1.2 不同数制之间的转换

在数字通信和计算机系统中，往往使用二进制、八进制和十六进制数给计算机输入指令和数据，而人们习惯于用十进制来计数或运算，这就需要解决这些数制之间的转换。

1. 二进制数转换成十进制数

把二进制数转换为等值的十进制数，通常采用“加权法”。即，将任意一个二进制数写成它的按权展开式，然后将数码为 1 的那些位的权值按十进制相加，就可得到该二进制数的等值十进制数。

例 1-1 求二进制数 110101.1011 的等值十进制数。

解 该数的二进制按权展开式为：

$$(110101.1011)_2 = 1\times2^5+1\times2^4+0\times2^3+1\times2^2+0\times2^1$$
$$+1\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}+1\times2^{-4}$$
$$=32+16+0+4+0+1+0.5+0+0.125+0.0625$$
$$=53.6875$$

2. 十进制数转换成二进制数

十进制数转换成等值的二进制数，需要将十进制数的整数部分和小数部分分别进行转换，因两者的转换方法是不相同的。

（1）整数部分的转换

十进制整数转换为二进制整数，通常采用“除 2 取余”法。转换的步骤如下：

第一步：用二进制数的基数 2 除给定的十进制整数，所得的余数（0 或 1）即为所求二进制整数的最低位(K_0)。

第二步：再用 2 除第一步所得的商，得余数（0 或 1）为所求二进制整数的次低位(K_1)。

第三步：重复用 2 除前一步所得的商，得余数（0 或 1），一直进行到商数得 0 为止，末次所得的余数为所求二进制数的最高位。

例 1-2 将十进制数 178 转换为等值的二进制数。

解

		余数
2	178	
2	89	0 (K_0)
2	44	1 (K_1)
2	22	0 (K_2)
2	11	0 (K_3)
2	5	1 (K_4)
2	2	1 (K_5)
2	1	0 (K_6)
	0	1 (K_7)

于是得 $(178)_{10} = (10110010)_2$

（2）小数部分的转换

将十进制纯小数转换成二进制纯小数，通常采用“乘 2 取整”法。转换的步骤是：

第一步：用二进制数的基数 2 乘给定的十进制小数，所得乘积的整数（0 或 1）即为所求二进制小数的最高位(K_{-1})；

第二步：用 2 乘第一步所得乘积的小数部分，所得第二次乘积的整数部分（0 或 1），即为所求二进制小数的次高位(K_{-2})；

第三步：重复用 2 乘前一步所得乘积的小数部分，一直到所得乘积的小数部分为零，或达到转换精度为止。

例 1-3 将十进制小数 0.6875 转换成等值的二进制数。

解

	乘积整数部分
$0.6875 \times 2 = 1.375$	1 (K_{-1})
$0.375 \times 2 = 0.75$	0 (K_{-2})
$0.75 \times 2 = 1.50$	1 (K_{-3})
$0.50 \times 2 = 1.0$	1 (K_{-4})

于是得 $(0.6875)_{10}=(.1011)_2$

带小数点的任意十进制数转换为等值的二进制数，可分别运用上述方法，将十进制数的整数部分和小数部分，分别转换成相应的二进制数，再将所得的二进制数的整数和小数相加，即可得到所转换的等值二进制数。

例 1-4 将十进制数 19.625 转换成等值二进制数。

解 ① 整数部分用“除 2 取余”法进行转换

2	19	余数
2	9	1 (K_0)
2	4	1 (K_1)
2	2	0 (K_2)
2	1	0 (K_3)
	0	1 (K_4)

故 $(19)_{10}=(10011)_2$

② 小数部分用“乘 2 取整”法进行转换。

$0.625\times2=1.25$ 整数部分为 1 (K_{-1})

$0.25\times2=0.50$ 整数部分为 0 (K_{-2})

$0.50\times2=1.0$ 整数部分为 1 (K_{-3})

故 $(0.625)_{10}=(101)_2$

③ 将整数部分和小数部分相加，得转换结果为：

$$(19.625)_{10}=(10011.101)_2$$

最后，需要指出的是，不是所有的十进制小数都能转换成有限位的二进制小数，因为用 2 去乘乘积的小数部分，若小数部分不为 0，则演算将一直进行下去。在用计算机进行转换的情况下，由于计算机的字长(位数)有限，当演算到最低位时，往往采用“四舍五入”的办法处理，这将出现误差，其误差的大小(或称转换的精度)与计算机的字长(位数)有关。

3. 二进制数与八进制、十六进制数的相互转换

由于八进制数的基数 $8=2^3$，而十六进制数的基数 $16=2^4$，所以三位二进制数恰好等于一位八进制数，四位二进制数恰好等于一位十六进制数，因此，二进制数转换成等值的八进制数(或十六进制数)的规则是：

从二进制数的小数点处开始，向左右两边按每三位(或四位)二进制数划为一组，不是三位(或四位)的，整数部分可在最高位的左边添 0，小数部分可在最低位的右边添 0，每组用一位等值的八进制(或十六进制)数代替，即可得到相应的八进制(或十六进制)数，现举例说明如下：

例 1-5 将二进制数 $(10101100001.11101111)_2$ 转换成等值的八进制数和十六进制数。

解 $(10101100001.11101111)_2=$ 010/101/100/001/.111/011/110 $=(2541.736)_8$

$(10101100001.11101111)_2=$ 0101/0110/0001/.1110/1111 $=(561.EF)_{16}$

八进制（或十六进制）数，转换成等值的二进制数时，只要按上述规则进行逆变换即可，

请读者自行练习，此处不再赘述。

几种常用数制之间的变换关系如表 1-1 所示。

表 1-1

十进制数	二进制数	八进制数	十六进制数
0	0 0 0 0 0	0	0
1	0 0 0 0 1	1	1
2	0 0 0 1 0	2	2
3	0 0 0 1 1	3	3
4	0 0 1 0 0	4	4
5	0 0 1 0 1	5	5
6	0 0 1 1 0	6	6
7	0 0 1 1 1	7	7
8	0 1 0 0 0	10	8
9	0 1 0 0 1	11	9
10	0 1 0 1 0	12	A
11	0 1 0 1 1	13	B
12	0 1 1 0 0	14	C
13	0 1 1 0 1	15	D
14	0 1 1 1 0	16	E
15	0 1 1 1 1	17	F
16	1 0 0 0 0	20	10
17	1 0 0 0 1	21	11
18	1 0 0 1 0	22	12
19	1 0 0 1 1	23	13
20	1 0 1 0 0	24	14
21	1 0 1 0 1	25	15
22	1 0 1 1 0	26	16
23	1 0 1 1 1	27	17
24	1 1 0 0 0	30	18
25	1 1 0 0 1	31	19

1.2 编　码

在数字通信和计算机系统中，信息可分为数值信息和字符信息两大类。前一节我们已讨论了数值信息的表示方法。为了表示字符信息，往往也采用由若干位的二进制数码来表示，这种给每个信息所分配的二进制代码称为对信息的编码。在对信息编码的基础上，通常还附加一些二进制位来帮助检错或纠错，构成检错码或纠错码。

本节仅介绍一些常用的编码以及它们的特性，有关编码理论问题请参阅相关资料。

1.2.1 二－十进制编码

用特定的二进制码来代表每一个十进制数，即为二进制编码的十进制，简称二－十进制编码（Binary Coded Decimal Codes，缩写为 BCD 码）。

当用二进制代码表示 N 个不同信息时，所需的二进制代码的最少位数 n 应满足下列表达

式：

$$N \leqslant 2^n \tag{1-6}$$

一位十进制数，有0～9个不同的信息，$N=10$，故$n=4$，也就是说，至少需要四位二进制码才能表示一位十进制数。

用四位二进制码可以组成$2^4=16$个不同的二进制序列(或称码组)，用其中的十个码组分别代表十进制中0～9个数，剩下六个多余的码组，称为冗余码组。由于从十六个二进制码组中，任意选取十个码组的方案有很多种，因而产生了多种BCD码。下面着重讨论8421码、2421码和余3码三种常用的BCD码。它们的编码如表1-2所示。

表1-2

二进制码				代码对应的十进制数			
				自然二进制码	二一十进制码		
B_3	B_2	B_1	B_0		8421	2421	余3码
0	0	0	0	0	0	0	冗余码组
0	0	0	1	1	1	1	
0	0	1	0	2	2	2	
0	0	1	1	3	3	3	0
0	1	0	0	4	4	4	1
0	1	0	1	5	5	冗余码组	2
0	1	1	0	6	6		3
0	1	1	1	7	7		4
1	0	0	0	8	8		5
1	0	0	1	9	9		6
1	0	1	0	10	冗余码组		7
1	0	1	1	11		5	8
1	1	0	0	12		6	9
1	1	0	1	13		7	冗余码组
1	1	1	0	14		8	
1	1	1	1	15		9	

1. 8421 BCD码

从表1-2中可看出，8421 BCD码的编码特点是：十进制数0～9所对应的四位二进制代码，就是与该十进制数等值的二进制数，即四位二进制数0000～1111十六种码组中的前十种码组0000～1001。后面的六个码组1010～1111在8421 BCD码中是不允许出现的，称冗余码组。8421码各位的权值从左到右分别是8、4、2、1，所以称这种编码为8421 BCD码。

8421 BCD码是一种有权码，有权码的数值与代码之间的关系（即它所代表的十进制数）应满足下列关系式：

$$N = \sum_{i=0}^{3} W_i a_i \tag{1-7}$$

式中N代表数值；W_i是第i位的权值；a_i是第i位上二进制的数码(0或1)。

例如 8421 码的 0111 所代表的十进制数为：

$$N=0\times8+1\times4+1\times2+1\times1=(7)_{10}$$

8421 码具有奇偶特性，即：凡是奇数(1,3,5,7,9)码组的最低位二进制数皆为 1，而偶数(0,2,4,6,8)码组的最低位二进制数皆为 0，因此，采用 8421 码容易判别它的奇偶性。

用 8421 BCD 码对一个多位十进制数进行编码，只要把十进制数的各位数字编成对应的 8421 码即可。例如，十进制数 579 的 8421 BCD 编码为：

$$(579)_{10}=(010101111001)_{8421\ BCD}$$

2. 2421 BCD 码

从表 1-2 可看到，2421 BCD 码的编码特点是：从十六个四位二进制码组中首尾各取五个码组，用前五个码组（0000～0100）分别表示十进制数 0～4；用后五个码组（1011～1111）分别表示十进制数 5～9。

2421 BCD 码也是一种有权码，所以它的数值与代码之间的关系也满足式（1-7），不过，它的权值分别是 2,4,2,1。例如，2421 BCD 码 1101 表示的十进制数是：

$$N=1\times2+1\times4+0\times2+1\times1=(7)_{10}$$

2421 BCD 码是一种自补码，即，在它的十个数码中，0 和 9、1 和 8、2 和 7、3 和 6、4 和 5 互为反码。例如数码 4 的 2421 BCD 的编码是 0100，而数码 5 的 2421 BCD 的编码是 1011，恰好是 0100 的反码。我们把具有这种特性的 BCD 码，称为自补码。

自补码的优点是求补简便，所以 2421 BCD 码在计算机系统中得到了广泛的使用。

应该指出的是，2421 BCD 的编码方案不是唯一的。因为在 2421 码中，有部分数码存在两种加权方法。例如，十进制数 7，既可编成 1101 也可编成 0111。用 2421 BCD 码对多位十进制数进行编码的方法与用 8421 码编码方法相同，

3. 余 3 BCD 码

从表 1-2 中可看到，余 3 BCD 码的编码特点是：它所表示的十进制数 0～9 的代码数值等于该代码各位的加权数之和再加常数“3”。例如十进制数 9 的 8421 代码是 1001，则余 3 BCD 代码应为

$$1001+0011=1100$$

余 3 BCD 码是一种特殊的有权码。因为，在余 3 BCD 码的码组中，将代码为“1”的那些位的权值之和，与它所代表的十进制数相差一个固定的常数，所以又称它为余权码。余权码的数值与它所表示的十进制数应满足下列关系式：

$$N=\sum_{i=0}^{3}W_ia_i+c \tag{1-8}$$

式中 c 为某一常数，对余 3 码来说，$c=-3$

余 3 BCD 码，也是一种自补码，因为，它的 0～4 的代码分别与 9～5 的代码互为反码，所以余 3 码在计算机系统中也得到广泛的应用。

1.2.2 循环码

循环码是一种常用的单位距离码。它的编码特点是：任何一个相邻码组(包括首尾两个码组)之间仅能有一个码位不同。例如，十进制数 6 和 7 是相邻的两个码组，它们的循环码分别为 0101 和 0100，这两个码组的编码仅在最低码位上不同，其他各码位均相同。

循环码的这一单位距离特性，能避免在码组转换的过渡过程中产生瞬时误码。因此循环码在通信和测量技术中得到了广泛的应用。四位循环码的编码如表 1-3 所示。

表 1-3

十进制数	循环码	十进制数	余 3 循环码
0	0 0 0 0		0 0 0 0
1	0 0 0 1	冗余码组	0 0 0 1
2	0 0 1 1		0 0 1 1
3	0 0 1 0	0	0 0 1 0
4	0 1 1 0	1	0 1 1 0
5	0 1 1 1	2	0 1 1 1
6	0 1 0 1	3	0 1 0 1
7	0 1 0 0	4	0 1 0 0
8	1 1 0 0	5	1 1 0 0
9	1 1 0 1	6	1 1 0 1
10	1 1 1 1	7	1 1 1 1
11	1 1 1 0	8	1 1 1 0
12	1 0 1 0	9	1 0 1 0
13	1 0 1 1		1 0 1 1
14	1 0 0 1	冗余码组	1 0 0 1
15	1 0 0 0		1 0 0 0

由表 1-3 还可以看出，循环码具有反射特性，即它的最高位的 0 和 1 只改变一次，若在最高位 0 和 1 的交界处设一镜面（表中虚线所示），则其他各位的编码以镜面为界形成镜像反射。

用循环码来表示十进制数时，为使 0 和 9 的码组也只能有一个码位不同（单位距离码），常采用如表 1-3 所示的余 3 循环码。它是从四位循环码的十六个码组中去除首尾各三个码组而构成的。

分析余 3 循环码的编码特点，可知它也具有反射特性。

循环码的各位没有确定的权值，它属于无权码，无权码得不到它的按权展开式。所以用循环码作运算时，必须首先将它转换成二进制数。下面就来讨论循环码与二进制数之间相互转换的方法。

1. 循环码转换成二进制数

由于在转换中要运用模二加运算，所以下面首先给出模二加的运算规则：

$$0 \oplus 0 = 0$$

$$0 \oplus 1 = 1$$

$$1 \oplus 0 = 1$$

$$1 \oplus 1 = 0 \quad \text{（不考虑进位）}$$

式中$\oplus$是模二加的运算符（模二加的运算就是后面要讨论的异或逻辑运算）。

循环码转换成相应的二进制码可按下列表达式求得：

$$B_i = B_{i+1} \oplus G_i \tag{1-9}$$