北京市高等教育精品教材立项项目

高等学校教材

数字电子技术基础

侯建军　主编
熊华钢　路而红　张晓冬　编

高等教育出版社

北京市高等教育精品教材立项项目

高等学校教材

数字电子技术基础

侯建军　主编
熊华钢　路而红　张晓冬　编

高等教育出版社

内容简介

本书以数字逻辑为基础，系统分析为桥梁，系统综合为目的，全面介绍数字逻辑的基本理论、分析方法、综合方法和实际应用。本书共分九章，第一章介绍数字逻辑的表示方法、布尔代数以及逻辑化简的基本方法；第二至五章分别讨论典型集成电路的基本工作原理及外特性、组合及时序电路的分析、设计方法和各种中规模逻辑模块的应用；第六章介绍典型中、大规模集成电路，高密度可编程逻辑器件及实用可编程门阵列的原理、组成，同时介绍了应用这些元件实现数字电路的方法；第七、八章介绍 A/D 和 D/A 转换器和脉冲电路；最后一章介绍数字系统设计方法，并给出了数字系统设计实例。

本书 2001 年被列为北京市高等教育精品教材重点立项项目，2003 年被列为高等教育百门精品课程教材建设计划五项研究项目。教材内容新颖、概念清楚、实践性强，在体现科学性、先进性和系统性方面具有特色。此外，书中附有大量图表和应用实例，便于自学，章末附有自我检测、思考题和习题，利于读者巩固和综合运用所学知识。

本书可作为高等学校通信、控制、电气、电子信息和计算机等专业的大学本科教材，同时也是从事电路设计、通信工程及计算机等专业的广大科技工作者参考用书。

图书在版编目(CIP)数据

数字电子技术基础/侯建军主编. —北京：高等教育出版社，2003.12
ISBN 7-04-013026-2

Ⅰ.数… Ⅱ.侯… 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(2003)第 099599 号

出版发行	高等教育出版社	购书热线	010-64054588
社　　址	北京市西城区德外大街 4 号	免费咨询	800-810-0598
邮政编码	100011	网　　址	http://www.hep.edu.cn
总　　机	010-82028899		http://www.hep.com.cn
经　　销	新华书店北京发行所		
印　　刷	国防工业出版社印刷厂		
开　　本	787×960　1/16	版　　次	2003 年 12 月第 1 版
印　　张	30.75	印　　次	2003 年 12 月第 1 次印刷
字　　数	570 000	定　　价	35.00 元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换。

前　言

本教材于2001年被列为北京市高等教育精品教材立项项目，2003年被列为高等教育百门精品课程教材建设计划立项研究项目，是教育部高等教育教学改革项目“电工电子系列课程教学内容和课程体系改革的研究与实践”的成果之一。书中基本内容符合原国家教育委员会颁布的课程教学基本要求，同时，力求反映本学科正在发展或已趋于成熟的内容。

在编写时，充分吸收新概念、新理论和新技术，力求处理好先进性和适用性的关系、处理好教材内容变化和基础内容相对稳定的关系。教材内容力求重点突出，基本概念明确清晰，贯穿少而精和理论联系实际的精神。

编写过程中，参考了国内外优秀教材，并总结近年来教学体会，增加了应用VHDL设计集成电路的现代设计方法，并有机融入各章节之中，同时与传统设计方法进行了充分地比较。另外，增加了自我检测、思考题，在习题中加入了EDA练习题，帮助学生加深对课程内容的理解，部分习题有一定的深度，以使学生在深入掌握课程内容的基础上扩展知识。

本教材配备教师授课使用的电子教案。教案图文并茂，教师可以根据授课情况组织各种教学信息，加深学生对课程内容的理解。

参加本书编写的教师多年从事电子电路课程体系、课程内容的改革，总结了多年的教学经验。本书由侯建军教授任主编，熊华钢教授任副主编。第一、三章由熊华钢教授执笔，第二、四章由路而红教授执笔，第七、八章由张晓冬教授执笔。侯建军执笔其余各章，并对全书进行了整理和统稿。万磊老师编写了第五、六、七、八、九章电子教案。娄淑琴、路勇和曾涛老师也给予了许多帮助。黄正瑾教授不辞辛苦地认真审阅了全部书稿，并提出了许多宝贵意见。借此机会也向所有关心、支持和帮助过本书编写、修改、出版、发行工作的同志们致以诚挚的谢意。

限于水平，书中难免出现不妥之处及错误，恳请读者批评指正。

编　者

2003年8月

策划编辑　张培东
责任编辑　李　刚
封面设计　李卫青
责任绘图　尹文军
版式设计　张　岚
责任校对　康晓燕
责任印制　杨　明

目　录

MAG 5-3/07

第一章

数字逻辑基础

电子系统中的信号可以分为两大类：模拟信号和数字信号。模拟信号是时间连续、数值也连续的信号。模拟信号来自于自然界客观存在的一些物理量，例如，速度、温度、压力和声音等。处理模拟信号的电路称为模拟电路。数字信号是在时间上和数值上均离散的信号，例如电子表给出的时间信号、生产流水线上记录零件个数的计数信号等。处理数字信号的电路称为数字电路。当强调处理电路功能的完整性和实用性时，也称其为数字系统。

数字系统已经成为人们日常生活中的重要组成部分。计算机、电话系统、高清晰度电视、光盘播放机、家电控制系统等等都是数字系统的例子。由于数字信号便于存储、处理和传输，数字化已成为当今电子技术的发展潮流。

本章首先讨论数字系统中数的表示方法及常用的几种编码，然后介绍逻辑代数的基本概念和基本理论，说明逻辑函数的基本表示形式及其化简。本章的内容为读者学习和掌握数字电路的分析和设计奠定了基本数学基础。

第一节　数制与编码

一、数制

数制是计数的方法，通常按一定的进位方式计数，称为进位计数制。在日常生活中，人们习惯使用十进制，然而在数字电路中常使用的是二进制，有时也使用八进制或十六进制。

（一）十进制（Decimal）

十进制数由0～9十个不同的数字符号（也称为数码）和一个小数点符号“.”组成，其计数规律为“逢十进一”。可见，十进制是以10为基数的进位计数制。每一个处在不同数位上的数码所代表的数值是不同的，不同的数位有不同的权值，从左至右由高位到低位排列。例如，十进制数652.5可表示为

$$(652.5)_D=6\times10^2+5\times10^1+2\times10^0+5\times10^{-1}$$

上式中等号左边的形式为进位制数的位置计数法，下标 D 表示十进制，也可以用下标 10 表示十进制。右边的形式为该数的按权展开式。式中 10^2、10^1、10^0 和 10^{-1} 分别为百位、十位、个位和小数点右第一位的权值，也就是相应位的 1 所代表的实际数值。可见位数越高，权值越大，相邻高位的权值是低位权值的 10 倍。

任意一个十进制数 N 的位置计数法和按权展开式可表示为

$$(N)_D = (K_{n-1}K_{n-2}\cdots K_1K_0.K_{-1}\cdots K_{-m})_D = \sum_{i=-m}^{n-1} K_i 10^i \qquad (1.1.1)$$

式(1.1.1)中 n 和 m 为正整数，分别代表此十进制数的整数和小数部分的位数；K_i 代表第 i 位上的数码(0～9)；10^i 为第 i 位的权值。

十进制虽然是人们最习惯的计数体制，却难于用电路实现，因为使一个电路或者电子器件具有严格区分的十个状态来与十进制的十个不同数码一一对应，是比较困难的。因此在数字电路中一般不直接使用十进制。

（二）二进制（Binary）

二进制数只由 **0** 和 **1** 两个数码和一个小数点符号“.”组成，它同十进制数一样，从左至右由高位到低位排列。计数规律为“逢二进一”。因此，二进制是以 2 为基数的进位计数制。例如二进制数 **1011.101** 可表示为

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

式中 2^3、2^2、…、2^{-3} 分别为相应位的权值，相邻高位的权值是低位权值的 2 倍。下标“B”表示二进制，也可以用下标“2”表示二进制。

二进制数只有两个数字符号 **0** 和 **1**，因此很容易用电路器件的状态来表示。例如，三极管的截止和饱和、继电器的接通和断开、电平的高和低等，都可以将其中一个状态规定为 **0**，另一个状态规定为 **1**，用来表示二进制数。这种表示方法简单方便，处理、存储和传输数据也十分可靠。

（三）十六进制（Hexadecimal）与八进制(Octal)

由于二进制数比十进制数位数多，不便于书写和记忆，因此常用十六进制数或八进制数来表示二进制数。

由十进制和二进制的介绍可推断出，任意 R 进制数由 0～(R−1)共 R 个不同的数码和一个小数点符号“.”组成，基数为 R，计数规律为“逢 R 进一”，每一个数位均有固定的权值 R^i，相邻高位的权值是低位权值的 R 倍。

任意 R 进制数 N 可表示为

$$(N)_R = (K_{n-1}K_{n-2}\cdots K_1K_0.K_{-1}\cdots K_{-m})_R = \sum_{i=-m}^{n-1} K_i R^i \qquad (1.1.2)$$

式(1.1.2)中 n 和 m 为正整数，分别代表 R 进制数的整数和小数部分的位数；K_i 代表第 i 位上的数码(0～$R-1$)；R^i 为第 i 位的权值。

用 16 或者 8 代替 R 作为基数，参照式(1.1.2)不难得到十六进制数和八进制数的表示方法。下标 H 或者 16 表示十六进制，下标 O 或者 8 表示八进制。需要注意的是十六进制数的数码由 0、1、2、…、9 和 A、B、C、D、E、F 十六个字符表示。

容易写出十六进制数和八进制数的按权展开式，如

$$(\mathrm{F8C.B})_{\mathrm{H}}=\mathrm{F}\times16^2+8\times16^1+\mathrm{C}\times16^0+\mathrm{B}\times16^{-1}$$

$$(7016.5)_{\mathrm{O}}=7\times8^3+0\times8^2+1\times8^1+6\times8^0+5\times8^{-1}$$

表 1.1.1 列出了几种常用数制的对照表。

表 1.1.1　几种常用数制对照表

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

二、不同数制之间的转换

数制转换就是一个数从一种进位制表示形式转换成等值的另一种进位制表示形式，其实质为权值的转换。

（一）二进制转换成十进制

利用二进制数的按权展开式，可以将任意一个二进制数转换成相应的十进制数。

例 1.1.1　将二进制数 **10011.101** 转换成十进制数。

解：将每一位二进制数乘以该位的权值，然后相加，可得

$$(\mathbf{10011.101})_{\mathrm{B}}=\mathbf{1}\times2^4+\mathbf{0}\times2^3+\mathbf{0}\times2^2+\mathbf{1}\times2^1+\mathbf{1}\times2^0+\mathbf{1}\times2^{-1}+\mathbf{0}\times2^{-2}+\mathbf{1}\times2^{-3}=(19.625)_{\mathrm{D}}$$

同理，利用任意进制数的按权展开式，可以将一个任意进制数转换成相应的十进制数。

（二）十进制转换成二进制

通常使用基数乘除法将十进制数转换为二进制数，可用“除 2 取余”法将十

进制的整数部分转换成二进制数。任意一个十进制整数 N 可写成

$$(N)_D = K_n \times 2^n + K_{n-1} \times 2^{n-1} + K_{n-2} \times 2^{n-2} + \cdots + K_1 \times 2^1 + K_0 \times 2^0$$

式中 K_n、K_{n-1}、…、K_1、K_0 是等值的二进制数各位数码。将等式两边分别除以 2，得

$$(N)_D/2 = K_n \times 2^{n-1} + K_{n-1} \times 2^{n-2} + K_{n-2} \times 2^{n-3} + \cdots + K_1 \times 2^0 + K_0/2$$

可以看出，十进制数 N 除以 2，余数为 K_0。将上式的商再除以 2，得

$$K_n \times 2^{n-2} + K_{n-1} \times 2^{n-3} + K_{n-2} \times 2^{n-4} + \cdots + K_2 \times 2^0 + K_1/2$$

余数为 K_1。依次类推，反复将每次得到的商再除以 2，直到商为 0，就可根据余数得到二进制数的每一位数码。这种将十进制的整数部分转换成二进制数的方法称为“除 2 取余”法。

例 1.1.2　将十进制数 29 转换成二进制数。

解：根据“除 2 取余”法，按如下步骤转换：

$$\begin{array}{ccccccccc} & \div 2 & & \div 2 & & \div 2 & & \div 2 & \div 2 \\ 0 & \leftarrow & 1 & \leftarrow & 3 & \leftarrow & 7 & \leftarrow 14 & \leftarrow 29 \\ & \downarrow & & \downarrow & & \downarrow & & \downarrow & \downarrow \\ & \mathbf{1} & & \mathbf{1} & & \mathbf{1} & & \mathbf{0} & \mathbf{1} \\ & K_4 & & K_3 & & K_2 & & K_1 & K_0 \end{array}$$

余数

则 $(29)_D = (\mathbf{11101})_B$。

可用“乘 2 取整”的方法将十进制数的纯小数部分转换成二进制数。任意十进制纯小数 N 可以写成

$$(N)_D = K_{-1} \times 2^{-1} + K_{-2} \times 2^{-2} + \cdots + K_{-(n-1)} \times 2^{-(n-1)} + K_{-n} \times 2^{-n}$$

式中 K_{-1}、K_{-2}、…、$K_{-(n-1)}$、K_{-n} 是二进制小数的各位数码。将等式两边分别乘以 2，得

$$2 \times (N)_D = K_{-1} + K_{-2} \times 2^{-1} + \cdots + K_{-(n-1)} \times 2^{-(n-2)} + K_{-n} \times 2^{-(n-1)}$$

由上式可以看出，十进制小数乘以 2，得到整数为 K_{-1}。

同理，将上面乘积的小数部分再乘以 2，得到的整数为 K_{-2}。依次类推，反复将每次得到的乘积的小数部分再乘以 2，直到小数部分为 0，或者小数部分虽不为 0，但二进制小数的位数或精度已达到要求，就可得到二进制小数的每一位数码。这种将十进制的小数部分转换成二进制数的方法称为“乘 2 取整”法。

例 1.1.3　将十进制数 $(0.723)_D$ 转换成误差 ϵ 不大于 2^{-6} 的二进制数。

解：用“乘 2 取整”法，按如下步骤转换

$$
\begin{array}{cccccccccccc}
 & \times 2 & & \times 2 & & \times 2 & & \times 2 & & \times 2 & & \times 2 \\
0.723 & \rightarrow & 0.446 & \rightarrow & 0.892 & \rightarrow & 0.784 & \rightarrow & 0.568 & \rightarrow & 0.136 & \rightarrow & 0.272 \\
 & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
 & \mathbf{1} & & \mathbf{0} & & \mathbf{1} & & \mathbf{1} & & \mathbf{1} & & \mathbf{0} \\
 & K_{-1} & & K_{-2} & & K_{-3} & & K_{-4} & & K_{-5} & & K_{-6}
\end{array}
$$

取整

则有$(0.723)_{D}=(\mathbf{0.101110})_{B}$，转换误差$\varepsilon<2^{-6}$。

由于整数和小数的转换方法不同，任何十进制数可分别对其整数和小数部分按上述方法进行转换，然后再将两部分转换结果合并得到对应的二进制数。

把十进制数转换为其他任意 R 进制数的方法类似于十进制数转换为二进制数，即整数部分采用基数除法，小数部分采用基数乘法，不同之处在于基数不是 2 而是 R。

（三）二进制转换成十六进制

十六进制基数为 16，由于 $16=2^4$，4 位二进制数就相当于 1 位十六进制数。因此，可用“分组对应”法将二进制数转换为十六进制数。

将二进制数转换成十六进制数时，首先从小数点开始，将二进制数的整数和小数部分每 4 位分为一组，不足 4 位的分别在整数的最高位前和小数的最低位后加 **0** 补足，然后每组用等值的十六进制码替代，即可得到所求的十六进制数。

例 1.1.4　将二进制数 **1011101.101001** 转换成十六进制数。

解：$(\mathbf{1011101.101001})_{B}=(\mathbf{0101\ 1101.1010\ 0100})_{B}=(5D.A4)_{H}$

同理，若将二进制数转换为八进制数，可将二进制数分为 3 位一组，再将每组的 3 位二进制数转换成 1 位八进制数即可。

（四）十六进制转换成二进制

由于每位十六进制数对应于 4 位二进制数，十六进制数转换成二进制数时，按位的高低依次排列将每 1 位十六进制数变成 4 位二进制数。

例 1.1.5　将十六进制数 B6.3A 转换成二进制数。

解：$(B6.3A)_{H}=(\mathbf{1011\quad 0110.0011\quad 1010})_{B}$

同理，若将八进制数转换为二进制数，按位的高低依次排列将每 1 位八进制数变成 3 位二进制数。

三、二进制正负数的表示及运算

（一）二进制原码、补码及反码

各种数制都有原码和补码之分。前面介绍的十进制和二进制数都属于原码。补码分为两种：一种称为基数的补码；另一种称为降基数的补码。这里仅讨论二进制数原码及补码表示法。

二进制数 N 的基数的补码又称为 2 的补码，常简称为补码，其定义为

$$[N]_{补} = 2^n - N \tag{1.1.3}$$

式(1.1.3)中，n 是二进制数 N 整数部分的位数。例如

$$[\mathbf{1010}]_{补} = 2^4 - \mathbf{1010} = \mathbf{10000} - \mathbf{1010} = \mathbf{0110}$$

$$[\mathbf{1010.101}]_{补} = 2^4 - \mathbf{1010.101} = \mathbf{10000.000} - \mathbf{1010.101} = \mathbf{0101.011}$$

二进制数 N 的降基数补码又称为 1 的补码，习惯上称为反码，其定义为

$$[N]_{反} = (2^n - 2^{-m}) - N \tag{1.1.4}$$

式(1.1.4)中，n 是二进制数 N 整数部分的位数，m 是 N 的小数部分的位数。例如

$$[\mathbf{1010}]_{反} = (2^4 - 2^0) - \mathbf{1010} = \mathbf{1111} - \mathbf{1010} = \mathbf{0101}$$

$$[\mathbf{1010.101}]_{反} = (2^4 - 2^{-3}) - \mathbf{1010.101} = \mathbf{1111.111} - \mathbf{1010.101} = \mathbf{0101.010}$$

从上述例子可以看到，在求二进制数的反码时，可对该数逐位求反得到，这也是二进制数的降基数补码称为反码的原因。

根据定义，二进制数的补码可由反码在最低有效位加 1 得到。例如

$$N = \mathbf{10110110}$$

$$[N]_{反} = \mathbf{01001001}$$

$$[N]_{补} = \mathbf{01001010}$$

无论是补码还是反码，按定义再求补或求反一次，将还原为原码。

（二）二进制正负数的表示法

带符号的十进制数，如+54、−33.6 等，由符号和绝对值两部分组成。但是数字电路不识别“+”和“−”符号。因此常用二进制数码的最高位来表示正、负号，用 **0** 表示正，**1** 表示负，其余各位表示数的绝对值，称为数值位，两部分合起来构成带符号的二进制数。

二进制正负数的表示法有原码、反码和补码三种表示方法。对于正数而言，三种表示法都是一样的，即符号位为 **0**，随后是二进制数的绝对值，即原码。例如

$$(+43)_D = \mathbf{00101011}$$

这里码长为 1 字节(即 8 位二进制位)，首位是符号位，后面 7 位是数的绝对值。

二进制负数的原码、反码和补码三种表示方法分别为符号位 **1** 加原码、符号位 **1** 加反码、符号位 **1** 加补码。

例 1.1.6　分别写出−25 的二进制原码、反码和补码，设码长为 1 字节。

解：码长为 8 位，符号位占 1 位，数值位占 7 位。25 的 7 位二进制原码为 **0011001**，对应的反码为 **1100110**，补码为 **1100111**。根据二进制负数的原码、反码和补码的定义有

$$[-25]_{原} = \mathbf{10011001}, [-25]_{反} = \mathbf{11100110}, [-25]_{补} = \mathbf{11100111}$$

表 1.1.2 列出了 4 位二进制数的三种表示方法，由表 1.1.2 可以看出，n 位带符号二进制数可以表示的数值范围是

$$\text{原码：}-(2^{n-1}-1)\sim+(2^{n-1}-1)$$

$$\text{反码：}-(2^{n-1}-1)\sim+(2^{n-1}-1)$$

$$\text{补码：}-2^{n-1}\sim+(2^{n-1}-1)$$

表 1.1.2　4 位二进制带符号数的原码、反码和补码

十进制	二进制		
	原码	反码	补码
+8	—	—	—
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
−0	1000	1111	—
−1	1001	1110	1111
−2	1010	1101	1110
−3	1011	1100	1101
−4	1100	1011	1100
−5	1101	1010	1011
−6	1110	1001	1010
−7	1111	1000	1001
−8	—	—	1000

（三）补码的算术运算

在数字电路中，用原码运算求两个正数 M 和 N 的差值 $M-N$ 时，首先要对减数和被减数进行比较，然后由大数减去小数，最后决定差值的符号。完成这个运算的电路复杂，运算速度也很慢。如果用反码或补码实现减法运算，可把减法运算变成加法运算，即 $M-N$ 变为 $M+(-N)$，即被减数 M 加上减数 N 的反码或补码，这样就把原码减法运算变成了反码或补码加法运算。

1. 反码运算

反码在进行算术运算时不需判断两数符号位是否相同，两数反码之和等于两数之和的反码，即 $[X_1]_{反}+[X_2]_{反}=[X_1+X_2]_{反}$，符号位参加运算。当符号位有进位时需循环进位，即把符号位进位加到和的最低位。

例 1.1.7　已知 $X_1=$**0001000**，$X_2=-$**0000011**，求 X_1+X_2。

解：根据$[X_1]_{反}+[X_2]_{反}=[X_1+X_2]_{反}$，有

$$\begin{array}{rl} & [X_1]_{反}=\mathbf{0\ 0001000} \\ +) & [X_2]_{反}=\mathbf{1\ 1111100} \\ \hline & \boxed{1}\ \mathbf{0\ 0000100} \\ +) & \qquad\qquad\longrightarrow \mathbf{1}\quad \text{循环进位} \\ \hline & [X_1]_{反}+[X_2]_{反}=\mathbf{0\ 0000101} \end{array}$$

故得 $X_1+X_2=+\mathbf{0000101}$。

2. 补码运算

补码的运算与反码相似，两数补码之和等于两数之和的补码，即$[X_1]_{补}+[X_2]_{补}=[X_1+X_2]_{补}$，符号位参加运算。不过不需循环进位，如有进位，自动丢弃。

例 1.1.8　已知 $X_1=-\mathbf{0001000}$，$X_2=\mathbf{0001011}$，求 X_1+X_2。

解：根据$[X_1]_{补}+[X_2]_{补}=[X_1+X_2]_{补}$

$$\begin{array}{rl} & [X_1]_{补}=\mathbf{1\ 1111000} \\ +) & [X_2]_{补}=\mathbf{0\ 0001011} \\ \hline & \text{自动丢弃}\longleftarrow \boxed{1}\ \mathbf{0\ 0000011} \end{array}$$

可见$[X_1]_{补}+[X_2]_{补}=\mathbf{0\ 0000011}$，故得 $X_1+X_2=+\ \mathbf{0000011}$。

由于补码运算无循环进位，比反码运算简单，因而应用更为广泛。

值得注意的是，补码的运算应在其相应位数表示的数值范围内进行，否则将可能产生错误的计算结果。

四、常用的编码

用文字、符号或数码表示特定对象的过程称为编码。数字系统中常用的是二进制编码，就是用二进制代码表示有关对象。n 位二进制代码有 2^n 个状态，可以表示 2^n 个对象。下面介绍几种常用的二进制代码。

（一）二-十进制码

二-十进制码，又称 BCD 码(Binary-Coded-Decimal)，是用二进制代码来表示人们习惯的十进制数码的编码方法。

要用二进制代码来表示十进制的 0～9 十个数，至少要用 4 位二进制数。4 位二进制数有 16 种组合，可从这 16 种组合中选择 10 种组合分别来表示十进制的 0～9 十个数。选择方案有多种，这就形成了不同的 BCD 码。常用的 BCD 码见表 1.1.3。