21世纪高等学校规划教材

SHUZI DIANZI JISHU JICHU

数字电子技术基础

张星慧　齐　明　主　编
管金华　王凤丽　赵乐森　副主编

中国电力出版社
http://jc.cepp.com.cn

21世纪高等学校规划教材

SHUZI DIANZI JISHU JICHU

数字电子技术基础

主　编　张星慧　齐　明

副主编　管金华　王凤丽　赵乐森

主　审　姜　华

中国电力出版社

http://jc.cepp.com.cn

内 容 提 要

本书是根据教育部最新制定的高等院校电子技术课程教学基本要求，结合编者多年的教学实践，为进一步提高学生的综合素质与自主创新能力编写而成。本书语言精练，知识全面，深入浅出，通俗易懂。在保证理论知识够用的同时，注重理论联系实际，培养学生的各方面能力。

全书共分7章，分别是逻辑代数基础、逻辑门电路、组合逻辑电路分析与设计、触发器、时序逻辑电路分析与设计、脉冲信号的产生与整形电路、D/A和A/D转换电路。每章均编有经典例题和习题，章后配有相应的实验题目。教材最后附有习题答案。本教材总学时为54～64（不含实验）。

本书可作为高等院校应用型本科计算机、电子、通信、机电等专业的教材（高职院校可从中选取部分内容讲解），也可作为自学考试和从事电子技术工程人员的自学用书。

图书在版编目（CIP）数据

数字电子技术基础/张星慧，齐明主编. —北京：中国电力出版社，2010.10

21世纪高等学校规划教材

ISBN 978-7-5123-0838-1

Ⅰ.①数…　Ⅱ.①张…②齐…　Ⅲ.①数字电路－电子技术－高等学校－教材　Ⅳ.①TN79

中国版本图书馆CIP数据核字（2010）第175816号

中国电力出版社出版、发行
（北京三里河路6号　100044　http://jc.cepp.com.cn）
汇鑫印务有限公司印刷
各地新华书店经售
*
2010年11月第一版　　2010年11月北京第一次印刷
787毫米×1092毫米　16开本　18.75印张　455千字
印数0001－3000册　定价**30.00**元

前　言

数字电子技术是高等院校计算机、电子、通信、机电以及一些非电类专业的一门技术性基础课程。通过本课程的学习可以使学生获得数字电子技术的基本理论和基本技能。

本书在编写过程中，以应用型人才培养目标为依据，结合笔者多年工程实践经验，紧紧抓住该技术基础课程的特点，突出课程本身的基础性和实践性，给出了一些深入浅出的练习题目，理论与实践紧密结合，注重技能培养。我们编写的宗旨是：

(1) 以基本要求为依据，以够用、实用为尺度，对传统内容进行了处理，精简了不必要的理论讲解与推导，重点放在对知识应用性的介绍。

(2) 精选内容，主次分明，详略得当。

(3) 考虑到集成电路的发展及应用，本书大幅度压缩分立元件电路的设计和其他次要的内容，加强以集成电路为主的数字集成电路的功能和应用。

(4) 为了便于读者掌握数字电子技术的基础知识和电子电路的识图技能，本书在编写中以实用电路为主线，将必要的基础知识与电子实用电路融为一体，使读者能在识图中掌握电子技术的基础知识和电子器件的基本性能，并把其应用到电子电路的设计和制作中。

(5) 教材编写注意将培养学生能力的要求贯穿于整个教学中。本教材通过教学目标、教学要求以及例题、习题、实验等多种途径帮助学生建立本课程学习的正确思路，抓住重点，明确思路，真正从“应用”这个角度加强对知识的掌握。

本书由张星慧、齐明担任主编，管金华、王凤丽、赵乐森担任副主编。由姜华担任主审。此外，参加本书编写工作的还有解立明、盖君清、王维兰、刘均波、张虹、李秋潭、刘磊、王宇晓、纪官燚、郑建军、齐丽丽、张淑玲。

编写过程中，由于时间仓促，加之水平有限，书中错误和不妥之处在所难免，敬请各方面的读者予以批评指正，以便今后不断改进。

编　者

2009 年 7 月

目　录

第1章　逻辑代数基础

内容提要

(1) 常用数制及其转换；
(2) 码制及常用代码；
(3) 逻辑运算及逻辑函数的表示方法；
(4) 逻辑代数的基本定律和规则；
(5) 逻辑函数的公式化简法及卡诺图化简法。

1.1　数字电路概述

1.1.1　模拟信号和数字信号

在自然界中有形形色色的物理量，尽管它们的性质各异，但就其变化规律的特点而言，不外乎有两大类：模拟信号和数字信号。

1. 模拟信号

模拟信号是指时间和数值上都是连续变化的信号，它具有无穷多的数值，其数学表达式也较复杂，如正弦函数、指数函数等。如图 1-1 (a) 所示为典型的模拟信号。

人们从自然界感知的许多物理量均属于模拟性质的，如速度、压力、声音、温度等。在工程技术上，为了便于分析，常用传感器将模拟量转换为电流、电压或电阻等电量，以便用电路进行分析和处理。传输、处理模拟信号的电路称为模拟电子线路，简称模拟电路。在模拟电路中主要关心输入、输出信号间的大小、相位、失真等方面的问题。

2. 数字信号

电子系统中一般均含有模拟和数字两种构件。模拟电路是系统中必需的组成部分。但是，数字电路便于存储、分析或传输信号，更具优越性。

数字信号是指时间和数值上都不连续变化的信号，即数字信号具有离散性，如图 1-1 (b) 所示。交通信号灯控制电路、智力竞赛抢答电路，以及计算机键盘输入电路中的信号，

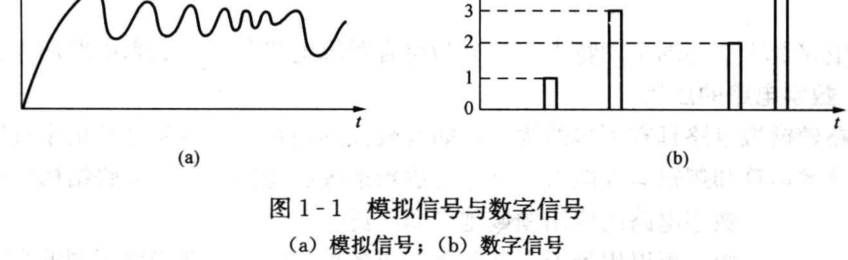

图 1-1　模拟信号与数字信号
(a) 模拟信号；(b) 数字信号

都是数字信号。对数字信号进行传输、处理的电子线路称为数字电子线路，简称数字电路。在数字电路中主要关心输入、输出之间的逻辑关系。

1.1.2 数字电路的特点

（1）数字电路中的工作信号是不连续的数字信号，反映在电路上只有高电平和低电平两种状态，因此在分析数字电路时采用二进制数码 0 和 1 来表示电路中的高、低两种电平状态。

（2）与模拟电路相同，数字电路也是由半导体器件如二极管、三极管、场效应管组成，但不同电路中器件的工作状态不同。数字电路在稳态情况下，半导体器件工作于开、关状态，这种开关状态是利用器件的导通和截止来实现的，器件的导通和截止反映在电路上就是电流的有无、电压的高低，这种有和无、高和低相对立的两种状态，正好可用二进制数码 0 和 1 来表示。因此，数字电路中的信号采用的是二进制表示，二进制数码 0 和 1 在此只代表两种不同的状态，没有数量的大小。例如，用 0 和 1 分别表示一件事的是与非、真与假，一盏灯的亮与灭，一个开关的开通与断开等。

（3）数字电路对元件的精度要求不高，允许有较大的误差，只要在工作时能够可靠地区分 0 和 1 两种状态就可以了。因此，数字电路便于集成化、系列化生产。它具有使用方便、可靠性高、价格低廉等特点。

（4）与模拟电路不同，数字电路讨论的是输入与输出之间抽象的逻辑关系，使用的主要方法是逻辑分析和逻辑设计，主要工具是逻辑代数，所以数字电路又称逻辑电路。

（5）数字电路能够对数字信号进行各种逻辑运算和算术运算，因此广泛应用于数控装置、智能仪表以及计算机中。

1.1.3 数字电路的分类

数字电路按其组成的结构不同可分为分立元件电路和集成电路两大类。分立元件电路是最基本的电路，它是由二极管、三极管、电阻、电容等元器件组成，并且所有元件都裸露在外，没有封装。随着集成电路的飞速发展，分立元件电路已逐步被取代，集成电路按集成度的大小分为小规模集成电路（SSI，集成度为 1～10 门/片）、中规模集成电路（MSI，集成度为 10～100 门/片）、大规模集成电路（LSI，集成度为 100～1000 门/片）、超大规模集成电路（VLSI，集成度大于 1000 门/片）。集成电路从应用的角度可分为通用型和专用型两大类，通用型是已被定型的标准化、系列化的产品，适用于不同的数字设备；专用型是指为某种特殊用途专门设计，具有特定的复杂而完整功能的功能块型产品，只适用于专用的数字设备。

数字电路按所用元器件的不同，可分为双极型和单极型电路。其中双极型电路又有 TTL、DTL、ECL、IIL、HTL 等多种；单极型电路有 JFET、NMOS、PMOS、CMOS 四种。

按数字电路逻辑功能的不同特点，又分为组合逻辑电路和时序逻辑电路两大类。

1.1.4 数字电路的应用

数字电路较模拟电路具有更多的优点，如有较强的稳定性、可靠性和抗干扰能力，精度较高，具有算术运算和逻辑运算能力，可进行逻辑推理和逻辑判断，电路结构简单，便于制造和集成等。因此，数字电路的应用领域越来越广泛。

在数字通信系统中，可以用若干个 0 和 1 编成各种代码，分别代表不同的含义，用以实

现信息的传送。

利用数字电路的逻辑推理和判断功能，可以设计出各式各样的数控装置，用来实现对生产和过程的自动控制。其工作过程是：首先用传感器在现场采集受控对象的数据，求出它们与设定数据的偏差，接着由数字电路进行计算、判断，然后产生相应的控制信号，驱动伺服装置对受控对象进行控制或调整。这样不仅能通过连续监控提高生产的安全性和自动化水平，同时也提高了产品的质量，降低了成本，减轻了劳动强度。

在数字电子技术基础上发展起来的数字电子计算机，是当代科学技术最杰出的成就之一。今天，电子计算机不仅成了近代自动控制系统中不可缺少的一个重要组成部分，而且已经渗透到了国民经济和人民生活的各个领域，成为人们工作、生活、学习不可或缺的重要组成部分，并在许多方面产生了根本性的变革。尤其是计算机网络技术的飞速发展，使人们获取信息、享受网络服务更为便捷。

然而，数字电路的应用也具有它的局限性。前面已提到，在自动控制和测量系统中，被控制和被测量的对象往往是一些连续变化的物理量，即模拟信号，而模拟信号不能直接为数字电路所接收，这就给数字电路的使用带来很大的不便。为了用数字电路处理这些模拟信号，必须用专门的电路将它们转换为数字信号（称为模—数转换）；而经数字电路分析、处理输出的数字量往往还要通过专门的电路转换成相应的模拟信号（称为数—模转换）才能为执行机构所接收。这样一来，不但导致了整个设备的复杂化，而且也使信号的精度受到影响，数字电路本身可以达到的高精度也因此失去了意义。因此，在使用数字电路时，应具体情况具体分析，以便于操作、提高生产效率为目的。

1.2 数　　制

数制即计数体制，它是按照一定规则表示数值大小的计数方法。日常生活中最常用的计数体制是十进制，数字电路中常用的是二进制，有时也采用八进制和十六进制。对于任何一个数，可以用不同的进制来表示。

1.2.1 各种数制

1. 十进制（Decimal）

十进制是我们最常使用的数制。在十进制中，共有 0～9 十个数码，所以它的运算规则是“逢十进一，借一当十”，故为十进制；同一数字符号在不同的数位代表的数值不同。设某十进制数 N_{10} 有 n 位整数，m 位小数，则可表示为

$$N_{10} = \sum_{i=-m}^{n-1} k_i 10^{i-1} \tag{1 - 1}$$

其中 k_i 为第 i 位的系数，可取 0，1，2，3，…，9；10^{i-1} 为第 i 位的权；10 为进位基数，基数和权是进位制的两个要素，利用基数和权，可以将任何一个数表示成多项式的形式。例如，十进制数 509.67 可表示成

$$(509.67)_{10} = 5\times10^2 + 0\times10^1 + 9\times10^0 + 6\times10^{-1} + 7\times10^{-2}$$

这种表示方法称为多项式表示法或按权展开式。

2. 二进制（Binary）

在数字电路中，应用最广的是二进制。二进制数中只有 0、1 两个数字符号，所以运算

规则是“逢二进一，借一当二”，各位的权为 2^{i-1}，k_i 为第 i 位的系数，设某二进制数 N_2 有 n 位整数，m 位小数，则可表示为

$$N_2 = \sum_{i=-m}^{n-1} k_i 2^{i-1} \tag{1-2}$$

利用式（1-2）可以将任何一个二进制数转换为十进制数。

【例 1-1】 将二进制数 1101.11 转换为十进制数。

解 $(1101.11)_2 = 1\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}=(13.75)_{10}$

计算机内采用的是二进制表示法，采用二进制具有以下优点：

（1）二进制只有 0 和 1 两个代码，因此，在数字系统中，可用电子器件的两种不同状态来表示这两个代码，实现起来非常方便。例如，用晶体管的导通和截止来表示 0 和 1，或用低电平和高电平来表示 0 和 1 等。所以，二进制数的物理实现简单、易行、可靠，并且存储和传送也方便。

（2）二进制运算规则简单，有利于简化计算机的内部结构，提高运算速度。

二进制数的缺点是书写位数太多，不便于记忆，为此数字系统通常用八进制和十六进制。

3. 八进制（Octal）

八进制有 0、1、2、3、4、5、6、7 八个数码，基数为 8，它的运算规则是“逢八进一，借一当八”。任意一个八进制数 N_8 可表示为

$$N_8 = \sum_{i=-m}^{n-1} k_i 8^{i-1} \tag{1-3}$$

利用式（1-3）可将任意一个八进制数转换为十进制数。

【例 1-2】 将八进制数 372.5 转换为十进制数。

解 $(372.5)_8 = 3\times8^2+7\times8^1+2\times8^0+5\times8^{-1}=(250.625)_{10}$

4. 十六进制（Hexadecimal）

十六进制数采用十六个数码，而且“逢十六进一，借一当十六”。这十六个数码是 0、1、2、3、4、5、6、7、8、9、A（对应于十进制数中的 10）、B(11)、C(12)、D(13)、E(14)、F(15)。十六进制数的基数是 16，仿照式（1-1），任一十六进制数 N_{16} 可表示为

$$N_{16} = \sum_{i=-m}^{n-1} k_i 16^{i-1} \tag{1-4}$$

利用式（1-4）可将任意一个十六进制数转换为十进制数。

【例 1-3】 将十六进制数 4E6. B7 转换为十进制数（小数取 3 位）。

解 $(4E6.B7)_{16} = 4\times16^2+14\times16^1+6\times16^0+11\times16^{-1}+7\times16^{-2}$

$\approx(1255.563)_{10}$

上面几个例题说明，当把二进制、八进制、十六进制数转换为十进制时，先按权展开为多项式，然后按十进制数进行计算，结果便是十进制数。在转换过程中要注意各位权的幂不要写错，系数为 0 的那些项可以不写。

十进制数（Decimal）、二进制数（Binary）、八进制数（Octal）、十六进制数（Hexadecimal）也常采用第一个字母 D、B、O、H 作为其标识，加在数的后面。

例如，$(F58.B2)_{16}$ 可写成 F58. B2H，还可写成 $(F58.B2)_H$。

二进制数与八进制数、十进制数、十六进制数之间的对应关系如表 1-1 所列。

表 1-1 二进制数与八进制数、十进制数、十六进制数之间的对应关系

十进制数	二进制数	八进制数	十六进制数	十进制数	二进制数	八进制数	十六进制数
0	0	0	0	9	1001	11	9
1	1	1	1	10	1010	12	A
2	10	2	2	11	1011	13	B
3	11	3	3	12	1100	14	C
4	100	4	4	13	1101	15	D
5	101	5	5	14	1110	16	E
6	110	6	6	15	1111	17	F
7	111	7	7	16	10000	20	10
8	1000	10	8				

1.2.2 数制转换

人们习惯用十进制，但送入机器时，必须将十进制数转换成数字系统能识别的二进制数。用二进制表示一个比较大的数时，位数较长不容易读写和记忆，这时常采用八进制和十六进制作为二进制的缩写。因此，务必要熟练掌握不同数制之间的转换。

1. 十进制数转换成非十进制数

(1) 十进制转换成二进制。

整数部分：用除2取余的方法进行转换，转换结果为“先余为低，后余为高”。

小数部分：用乘2取整的方法进行转换，转换结果为“先整为高，后整为低”。

【例1-4】 将 $(38.125)_{10}$ 转换成二进制。

解 整数部分

	余数	
2 | 38		
2 | 19	0	第一个余数为二进制数的最低位
2 | 9	1	
2 | 4	1	
2 | 2	0	
2 | 1	0	
0	1	最后一个余数为二进制数的最高位

小数部分

$0.125\times2=0.250$	取出整数 0	第一个整数为二进制数的最高位
$0.250\times2=0.50$	取出整数 0	
$0.50\times2=1.00$	取出整数 1	最后一个整数为二进制数的最低位

所以有：$(38.125)_{10}=(100110.001)_2$

【例1-5】 把十进制小数 0.39 转换成二进制小数。

①要求误差不大于 2^{-7}。

②要求误差不大于 0.2%。

解 ①要求误差不大于 2^{-7}，只需保留至小数点后7位，计算过程如下：

$$0.39\times2=0.78\cdots\cdots0$$

$$0.78\times2=1.56\cdots\cdots1$$
$$0.56\times2=1.12\cdots\cdots1$$
$$0.12\times2=0.24\cdots\cdots0$$
$$0.24\times2=0.48\cdots\cdots0$$
$$0.48\times2=0.96\cdots\cdots0$$
$$0.96\times2=1.92\cdots\cdots1$$

因此，$(0.39)_{10}\approx(0.0110001)_2$

②由于$\frac{1}{2^9}=\frac{1}{512}<0.2\%$，因此要求误差不大于0.2%，只需保留至小数点后9位。

接（1）的计算过程有：

$$0.92\times2=1.84\cdots\cdots1$$
$$0.84\times2=1.68\cdots\cdots1$$

因此，$(0.39)_{10}\approx(0.011000111)_2$

（2）十进制数转换成八进制。

十进制转换为八进制与十进制转换为二进制的方法类似。

整数部分：用除8取余的方法进行转换，转换结果为“先余为低，后余为高”。

小数部分：用乘8取整的方法进行转换，转换结果为“先整为高，后整为低”。

【例1-6】 将$(84.5)_{10}$转换成八进制。

解 整数部分

		余数	
8	84		
8	10	4	第一个余数为八进制数的最低位
8	1	2	↑
	0	1	最后一个余数为八进制数的最高位

小数部分

$0.5\times8=4.0$ 取出整数4，余数为0，转换结束。

综上可得：$(84.5)_{10}=(124.4)_8$

（3）十进制数转换成十六进制。

十进制转换为十六进制与十进制转换为二进制的方法类似。

整数部分：用除16取余的方法进行转换，转换结果为“先余为低，后余为高”；

小数部分：用乘16取整的方法进行转换，转换结果为“先整为高，后整为低”。

【例1-7】 将$(1002.45)_{10}$转换成十六进制。

解 整数部分

		余数	
16	1002		
16	62	10	第一个余数为十六进制数的最低位
16	3	14	↑
	0	3	最后一个余数为十六进制数的最高位

小数部分

$$0.45\times16=7.20\cdots\cdots7$$

$$0.20\times16=3.20\cdots\cdots3$$
$$0.20\times16=3.20\cdots\cdots3$$
$$0.20\times16=3.20\cdots\cdots3$$

得到：$(1002.45)_{10}\approx(3EA.7333)_{16}$

因此，转换误差校核为整数、小数两部分。又因为整数部分无误差，小数部分误差校核为：

$$(0.45)_{10}=7\times16^{-1}+3\times16^{-2}+3\times16^{-3}+3\times16^{-4}=0.449\,997$$

可以看出，从二进制、八进制、十六进制转换为十进制，或十进制整数转换为二进制，都能做到完全准确。但把十进制小数转换为其他进制小数时，除少数可以完全准确外，大多数会存在误差，这时就要根据精度的要求进行“四舍五入”。

2. 二进制数与八进制数之间的转换

由于1位八进制数有0～7八个数码，3位二进制数正好有000～111八种组合，它们之间有以下简单的对应关系

八进制数	0	1	2	3	4	5	6	7
二进制数	000	001	010	011	100	101	110	111

利用这种对应关系，可以很方便地在八进制数与二进制数之间进行转换。

将二进制数转换为八进制数的方法是：以小数点为界，将二进制数的整数部分从低位开始，小数部分从高位开始，每3位分成一组，头尾不足3位的补0，然后将每组3位二进制数转换为1位八进制数。

【例1-8】 将$(10111010011.01011)_2$转换成八进制。

解
$$\begin{array}{cccccccc}\underline{010} & \underline{111} & \underline{010} & \underline{011} & . & \underline{010} & \underline{110}\\ 2 & 7 & 2 & 3 & . & 2 & 4\end{array}$$

所以，$(101\backslash1010011.01011)_2=(2723.24)_8$

将八进制转换为二进制，只要将每1位八进制用3位二进制数表示即可。

【例1-9】 将$(3274.65)_8$转换成二进制。

解
$$\begin{array}{ccccccc}\underline{3} & \underline{2} & \underline{7} & \underline{4} & . & \underline{6} & \underline{5}\\ 011 & 010 & 111 & 100 & . & 110 & 101\end{array}$$

所以，$(3274.65)_8=(011010111100.110101)_2$

3. 二进制数与十六进制数之间的转换

由于1位十六进制数有16个代码，而4位二进制数正好有0000～1111十六种组合，它们之间也存在简单的对应关系。利用这种对应关系，可以很方便地在十六进制数与二进制数之间进行转换。转换方法与二、八进制数的转换类似，只是将二进制数中3位一组改为4位一组。

【例1-10】 将二进制数$(11010111010.011101)_2$转换成十六进制数。

解
$$\begin{array}{cccccc}\underline{0110} & \underline{1011} & \underline{1010} & . & \underline{0111} & \underline{0100}\\ 6 & B & A & . & 7 & 4\end{array}$$

所以，$(11010111010.011101)_2=(6BA.74)_{16}$

【例1-11】 将十六进制数$(94E.B0C)_{16}$转换成二进制数。

解
$$\begin{array}{ccccccc}\underline{9} & \underline{4} & \underline{E} & . & \underline{B} & \underline{0} & \underline{C}\\ 1001 & 0100 & 1110 & . & 1011 & 0000 & 1100\end{array}$$

所以，$(94E.B0C)_{16}=(100101001110.101100001100)_2$

4. 其他进制数转换成十进制数

将其他进制数转换成十进制数时，只要将该数写成按权展开式，然后将各项相加求出最终结果即可，此处不再赘述。

1.2.3 二进制正、负数的表示法

在十进制数中，可以在数字前面加上“+”、“－”号来表示正、负数，显然数字电路不能直接识别“+”、“－”号。因此，在数字电路中把一个数的最高位作为符号位，并用0表示“+”号，用1表示“－”号，像这样的符号也数码化的二进制数称为机器数。原来带有“+”、“－”号的数称为真值。例如：

十进制数	+67	－67
二进制数（真值）	+1000011	－1000011
计算机内（机器数）	01000011	11000011

通常，二进制正、负数（机器数）有3种表示方法：原码、反码和补码。

1. 原码

用首位表示数的符号，0表示正，1表示负，其他位则为数的真值的绝对值，这样表示的数就是数的原码。

【例1-12】 求$(+105)_{10}$和$(-105)_{10}$的原码。

解
$$[(+105)_{10}]_{原}=[(+1101001)_2]_{原}=(01101001)_2$$
$$[(-105)_{10}]_{原}=[(-1101001)_2]_{原}=(11101001)_2$$

0的原码有两种，即

$$[+0]_{原}=(00000000)_2$$
$$[-0]_{原}=(10000000)_2$$

原码简单易懂，与真值转换起来很方便。但若是两个异号的数相加或两个同号的数相减就要做减法，做减法就必须判别这两个数哪一个绝对值大，用绝对值大的数减去绝对值小的数，运算结果的符号就是绝对值大的那个数的符号，这样操作比较麻烦，运算的逻辑电路也较难实现。于是，为了将加法和减法运算统一成只做加法运算，就引入了反码和补码。

2. 反码

反码用得较少，它只是求补码的一种过渡。

正数的反码与其原码相同，负数的反码是这样求的：先求出该负数的原码，然后原码的符号位不变，其余各位按位取反，即0变1，1变0。

【例1-13】 求$(+65)_{10}$和$(-65)_{10}$的反码。

解 $[(+65)_{10}]_{原}=(01000001)_2$ $[(-65)_{10}]_{原}=(11000001)_2$

则 $[(+65)_{10}]_{反}=(01000001)_2$ $[(-65)_{10}]_{反}=(10111110)_2$

很容易验证：一个数的反码的反码就是这个数本身。

3. 补码

正数的补码与其原码相同，负数的补码是它的反码加1。

【例1-14】 求$(+63)_{10}$和$(-63)_{10}$的补码。

解 $[(+63)_{10}]_{原}=(00111111)_2$ $[(+63)_{10}]_{反}=(00111111)_2$

则 $$[(+63)_{10}]_{补}=(00111111)_2$$

$$[(-63)_{10}]_{原}=(10111111)_2 \qquad [(-63)_{10}]_{反}=(11000000)_2$$

则 $$[(-63)_{10}]_{补}=(11000001)_2$$

同样可以验证：一个数的补码的补码就是其原码。

引入了补码以后，两个数的加、减法运算就可以统一用加法运算来实现，此时两数的符号位也当成数值直接参加运算，并且有这样一个结论：两数和的补码等于两数补码的和。所以在数字系统中一般用补码来表示带符号的数。

【例 1-15】 用二进制补码运算求出 14＋10、14－10、－14＋10 和－14－10。

解 由于 14＋10 和－14－10 的绝对值为 24，所以必须用有效数字为 5 位的二进制数才能表示，再加上一位符号位，就得到 6 位的二进制补码。

根据前述计算补码的方法可知，＋14 的二进制补码应为 001110（最高位为符号位），－14的二进制补码为 110010，＋10 的二进制补码为 001010，－10 的二进制补码为 110110。计算结果分别为

$$\begin{array}{rr}
+14 & 0\ \ 01110 \\
+10 & 0\ \ 01010 \\
\hline
+24 & 0\ \ 11000
\end{array}
\qquad
\begin{array}{rr}
+14 & 0\ \ 01110 \\
-10 & 1\ \ 10110 \\
\hline
+4 & (1)\ 0\ \ 00100
\end{array}$$

$$\begin{array}{rr}
-14 & 1\ \ 10010 \\
+10 & 0\ \ 01010 \\
\hline
-4 & 1\ \ 11100
\end{array}
\qquad
\begin{array}{rr}
-14 & 1\ \ 10010 \\
-10 & 1\ \ 10110 \\
\hline
-24 & (1)\ 1\ \ 01000
\end{array}$$

从例 1-15 可以看出，若将两个加数的符号位和来自最高有效数字位的进位相加，得到的结果（舍弃产生的进位）就是和的符号。

需要指出的是，在两个同符号数相加时，它们的绝对值之和不可超过有效数字位所能表示的最大值，否则会得出错误的计算结果。

1.3 码制和常用代码

在数字设备中，任何数据和信息都要用二进制代码表示。二进制中只有两个符号：0 和 1。例如，对于 n 位二进制数，它有 2^n 种不同的组合，即可以代表 2^n 种不同的信息。指定用某一二进制代码组合去代表某一信息的过程叫编码。由于这种指定是任意的，所以存在多种多样的编码方案。本节介绍几种常用的编码。

1.3.1 二—十进制编码（BCD 码）

二—十进制编码是一种用 4 位二进制代码表示 1 位十进制数的编码，简称 BCD（Binary Coded Decimal）码。1 位十进制数有 0～9 十个数码，而 4 位二进制数有 16 种组态，指定其中的任意 10 种组态来表示十进制的 10 个数，因此 BCD 编码方案有很多，常用的有 8421 码、余 3 码、2421 码、5421 码等，如表 1-2 所列。

表 1-2 所列各种 BCD 码中，8421 码、2421 码和 5421 码都属于有权码，而余 3 码属于无权码。

表 1-2 几种常见的BCD码

十进制数 \ 编码种类	8421 码	余 3 码	2421 码	5421 码
0	0 0 0 0	0 0 1 1	0 0 0 0	0 0 0 0
1	0 0 0 1	0 1 0 0	0 0 0 1	0 0 0 1
2	0 0 1 0	0 1 0 1	0 0 1 0	0 0 1 0
3	0 0 1 1	0 1 1 0	0 0 1 1	0 0 1 1
4	0 1 0 0	0 1 1 1	0 1 0 0	0 1 0 0
5	0 1 0 1	1 0 0 0	1 0 1 1	1 0 0 0
6	0 1 1 0	1 0 0 1	1 1 0 0	1 0 0 1
7	0 1 1 1	1 0 1 0	1 1 0 1	1 0 1 0
8	1 0 0 0	1 0 1 1	1 1 1 0	1 0 1 1
9	1 0 0 1	1 1 0 0	1 1 1 1	1 1 0 0
权	8 4 2 1		2 4 2 1	5 4 2 1

1. 8421BCD 码

8421BCD 码是最常用的一种 BCD 码，它和自然二进制码的组成相似，4 位的权值从高到低依次是 8、4、2、1。但不同的是，它只选取了 4 位自然二进制码 16 个组合中的前 10 个组合，即 0000～1001，分别用来表示 0～9 十个十进制数，称为有效码，剩下的 6 个组合 1010～1111 没有采用，称为无效码。8421BCD 码与十进制数之间的转换只需直接按位转换即可。例如

$$(509.37)_{10} = (0101\quad 0000\quad 1001.0011\quad 0111)_{8421BCD}$$

$$(0111\quad 0100\quad 1000.0001\quad 0110)_{8421BCD} = (748.16)_{10}$$

2. 余 3 码

余 3 码由 8421 码加 3（0011）得到，或者说是选取了 4 位自然二进制码 16 个组合中的中间 10 个，而舍弃头、尾 3 个组合而形成。

余 3 码也常用于 BCD 码的运算电路中。若将两个余 3 码相加，其和将比所表示的十进制数及所对应的二进制数多 6，当和为 10 时，正好等于二进制数的 16，于是便从高位自动产生进位信号。一个十进制数用余 3 码表示时，只要按位表示成余 3 码即可。例如

$$(85.93)_{10} = (1011\quad 1000.1100\quad 0110)_{余3BCD}$$

3. 2421BCD 码和 5421BCD 码

2421BCD 码和 5421BCD 码都是有权码，从高位到低位的权值依次为：2、4、2、1 和 5、4、2、1，这两种码的编码方案都不是唯一的，表 1-2 中给出的是其中一种方案。

2421BCD 码在进行运算时，也具有和余 3 码类似的特点。

5421BCD 码较明显的一个特点是：最高位连续 5 个 0 后又连续 5 个 1。若计数器采用该种代码进行编码，在最高位可产生对称方波输出。

【例 1-16】 分别用 8421BCD 码、5421BCD 码、余 3 BCD 码代换下列十进制数。

(1) $(23)_{10}$；(2) $(605)_{10}$。

解 (1) $(23)_{10} = (0010\ 0011)_{8421BCD}$

$(23)_{10} = (0010\ 0011)_{5421BCD}$

$$(23)_{10} = (0101\ 0110)_{余3BCD}$$

(2) $(605)_{10} = (0110\ 0000\ 0101)_{8421BCD}$

$$(605)_{10} = (1001\ 0000\ 1000)_{5421BCD}$$

$$(605)_{10} = (1001\ 0011\ 1000)_{余3BCD}$$

1.3.2 可靠性编码

代码在产生和传输的过程中，难免会发生错误，为减少错误的发生，或者在发生错误时能迅速地发现和纠正，在工程应用中普遍采用了可靠性编码。利用该技术编出的代码叫可靠性代码，格雷码和奇偶校验码是其中最常用的两种。

1. 格雷码

格雷码有多种编码形式，但所有格雷码都有两个显著的特点：一是相邻性，二是循环性。相邻性是指任意两个相邻的代码间仅有 1 位的状态不同；循环性是指首尾的两个代码也具有相邻性。因此，格雷码也称循环码。表 1-3 列出了典型的格雷码与十进制码及二进制码的对应关系。

表 1-3　典型格雷码与十进制码及二进制码的对应关系

十进制码	二进制码	格雷码	十进制码	二进制码	格雷码
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

由于格雷码具有以上特点，因此时序电路中采用格雷码编码时，能防止波形出现“毛刺”，并可提高工作速度。这是因为，其他编码方法表示的数码，在递增或递减过程中可能发生多位数码的变化。例如，8421BCD 码表示的十进制数，从 7（0111）递增到 8（1000）时，4 位数码均发生了变化。但事实上数字电路（如计数器）的各位输出不可能完全同时变化，这样在变化过程中就可能出现其他代码，造成严重错误。如第 1 位先变为 1，然后再其他位变为 0，就会出现从 0111 变到 1111 的错误。而格雷码由于其任何两个代码（包括首尾两个）之间仅有 1 位状态不同，所以用格雷码表示的数在递增或递减过程中不易产生差错。

2. 奇偶校验码

数码在传输、处理过程中，难免发生一些错误，即有的 1 错成 0，有的 0 错成 1。奇偶校验码是一种能够检验出这种差错的可靠性编码。

奇偶校验码由信息位和校验位两部分组成，信息位是要传输的原始信息，校验位是根据规定算法求得并添加在信息位后的冗余位。奇偶校验码分奇校验和偶校验两种。以奇校验为例，校验位产生的规则是：若信息位中有奇数个 1，校验位为 0，若信息位中有偶数个 1，校验位为 1。偶校验正好相反。也就是说，通过调节校验位的 0 或 1 使传输出去的代码中 1 的个数恒为奇数或偶数。

接收方对收到的加有校验位的代码进行校验。信息位和校验位中 1 的个数的奇偶性符合