高等职业教育教材

电子类

数字电子技术

天津市高等教育自学考试委员会　组编

刘常澍　主编　　赵雅兴　主审

天津大学出版社

高等职业教育教材

数 字 电 子 技 术

天津市高等教育自学考试委员会　组编
刘常澍　主编
赵雅兴　主审

天 津 大 学 出 版 社

内容简介

本书是按照天津市考试中心拟定的大纲编写的。主要内容有数字逻辑基础、逻辑门电路、组合逻辑电路、集成触发器、时序逻辑电路、脉冲波形的产生与整形、存储器和可编程逻辑器件。

本书适合电子信息、通信、自动化等专业高等职业教育、自学考试、成人教育以及大专学生作为教材使用，也可供其他相关专业学生和教师选用。

图书在版编目(CIP)数据

数字电子技术/刘常澍编.—天津：天津大学出版社，2001.3

ISBN 7-5618-1414-3

Ⅰ.数…　Ⅱ.刘…　Ⅲ.数字电路-电子技术　Ⅳ.TN79

中国版本图书馆 CIP 数据核字(2001)第 06661 号

出版发行　天津大学出版社
出 版 人　杨风和
地　　址　天津市卫津路 92 号天津大学内(邮编:300072)
电　　话　发行部:022-27403647　邮购部:022-27402742
印　　刷　河北省昌黎县人民胶印厂
经　　销　全国各地新华书店
开　　本　787mm×1092mm　1/16
印　　张　15.5
字　　数　385 千
版　　次　2001 年 3 月第 1 版
印　　次　2001 年 3 月第 1 次
印　　数　1～3 000
定　　价　21.00 元

高等职业教育教材编审委员会名单

序

全国高等教育自学考试指导委员会副主任
中国职业技术教育学会副会长　王明达

中国高等教育大众化目标的实现必然伴随着高等教育形式和结构多样化的变革。单纯以学术水平为追求目标的高等教育无法满足社会对于多种专门人才的需求。因此要大力发展高等职业教育，培养社会需要的各类专门人才，以适应我国经济和社会发展的要求。

什么是高等职业教育？职业教育的特征不在于办学形式，主要体现在培养目标上。培养生产、服务、管理第一线的实用人才的教育即为职业教育。按照专业所需接受教育的年限达到相当于普通高等教育学习年限的职业教育即为高等职业教育。

高等职业教育如何实现培养实用人才的目标？首要的就是专业设置。既然培养的是生产第一线的实用人才，所设专业就一定是直接与社会生产、生活相联系的，是社会生产、生活中最必需的。这与普通高等学校开设专业的思路有着本质的区别。其次是教学内容的安排和教学计划的制定。接受高等职业教育的学生的学习内容必须是成熟的技术和管理规范，教学计划、课程设置应该按照职业岗位群的职业能力要求来确定，而不应从学科体系出发。再次，为使学生毕业就能基本顶岗工作，要求增大实习训练所占的比例，在校期间就基本完成上岗前的实践训练。为了保证实践训练得到社会认可，要实行学历证书与职业资格证书“双证书”制度，同时要求双师型教师任教。只有按部就班实现以上要求的高等职业教育才会被社会认同，也才会有生命力。

办出特色是高等职业教育生命力的源泉。学生毕业即能顶岗是职业教育区别于其他教育的一个突出特点。要想做到这一点，一方面学习理论知识要以必需和够用为度，让学生掌握基本理论和知识；一方面要多方面开辟实习基地，保证充足的实训时间。高等职业教育的水准主要是通过专业设置、课程内容以及实训能力的培养体现的。

为落实第三次全教会“完善自学考试制度、大力发展高等职业教育”的改革思路，1999 年全国高等教育自学考试指导委员会决定在天津市开展高等教育自学考试职业技术专业的试验工作。天津市高等教育自学考试委员会在深入调查研究

的基础上，从职业岗位群的技能需求出发，以能力本位教育（CBE）为理论依托，设计了12个职业技术专业，于2000年面向社会开考。

高等教育自学考试开考职业技术专业的试验在完善高等教育自学考试专业建设、拓展自学考试教育功能方面，在探索开放式教育培养应用型高级人才方面，在职业教育课程体系建设方面，在教育与产业的有机结合方面，在构建完整的职业教育体系方面，以及在实践技能考核的研究、管理方面对于我国高等教育自学考试制度的完善和高等职业教育的发展都具有重要意义。

天津市高等教育自学考试委员会将根据职业技术专业试验工作的需要陆续出版有关考试课程的教材。教材编撰者多为具有职业教育经验的学科专家和职业教育专家。他们根据职业教育的专业培养目标重新整合了学科知识体系，尽力体现理论知识必需、够用的原则。当然，由于认识水平的局限和时间的紧迫，这些教材还需要继续完善提高。尽管如此，迈出的这第一步是十分可贵的。我深信，高等教育自学考试职业技术专业的试验工作一定能取得成功。

2001年1月于北京

前　言

目前世界已进入信息时代。信息的处理、存储、传输要靠电子信息技术、计算机技术和网络通信技术，而这些技术都是以数字技术为基础的。数字系统和数字设备已广泛应用于各个领域，并且是必不可少的关键部分。因而电子信息技术、计算机技术以及相关技术领域的工程师和技术人员必须掌握数字系统的基础知识。

我国经济建设正以前所未有和世界少有的速度健康发展，需要大量各个层次的科技人才，因而各类教育迅猛发展。近年来不但普通高校的本、专科扩大招生规模，成人教育、高等职业教育、自学考试教育也蓬勃发展。然而各类教材显得不相适应。编写适应不同教育层次、不同读者对象的教材是教育工作者义不容辞的责任。

本书内容是按照天津市考试中心拟定的大纲组织编写的。它有别于本科教材，适合电子信息、通信、自动化等专业高等职业教育、自学考试、成人教育以及大专读者使用。本书基础理论深入浅出、实践性强，并备有大量例题和习题，因而也适合于自学。

全书内容有数字逻辑基础、逻辑门电路、组合逻辑电路、集成触发器、时序逻辑电路、脉冲波形的产生与整形、存储器和可编程逻辑器件。

书中一律采用国家标准图形符号，在出现符号的地方对其所表示的意义进行简要明确的解释。读者学习本书过程中应同时学会认读常用的逻辑符号。

本书由张子珍编写第 1 章和第 6 章，李春华编写第 3 章和第 7 章，袁秀英编写第 2 章，严新忠编写第 4 章，刘常澍编写第 5 章。赵雅兴教授对全部书稿进行了审阅，刘常澍对全书进行校订、统编。

在编写本书过程中一直受到天津市考试中心的指导、帮助，编者对考试中心同志的辛勤工作由衷感谢。

这里还要感谢杨淑琴、马欣、张耀娟为本书所做的工作。

限于编者的经验不足、水平有限，书中难免存在不妥乃至错误之处，敬请广大读者不吝赐教。

编者

2001 年 1 月于天津

目　录

第 1 章　数字逻辑基础

第 2 章　逻辑门电路

第 3 章　组合逻辑电路

第 4 章　集成触发器

第1章　数字逻辑基础

数字逻辑基础亦称逻辑代数基础。本章主要介绍描述数字电路逻辑功能的数学方法。首先简要介绍了数字电路中常用的数制与码制，然后介绍了逻辑代数的基本公式、常用公式和三个重要定理以及逻辑函数及其表示方法，最后着重讲述了逻辑函数的公式化简法和卡诺图化简法。

1.1　概述

1.1.1　数字量与模拟量

在自然界中，存在着许许多多的物理量，尽管它们的性质各异，但就其变化规律特点而言，不外乎两大类。

其中一类物理量在时间上和数量上的变化是离散的。也就是说，它们在时间上的变化是不连续的，总是发生在一系列离散的瞬间。同时，它们的数值大小和每次的增减变化都是某一个最小数量单位的整数倍，而小于这个最小数量单位的数值没有任何物理意义。这一类物理量叫做数字量，把表示数字量的信号叫做数字信号，工作在数字信号下的电子电路叫做数字电路。例如，用计数器记录从自动生产线上输出的部件数目时，每送出一个部件便给计数器一个信号，使之计1，而没有部件送出时加给计数器的信号是0，所以不计数。可见，部件数目这个信号无论在时间上还是在数量上都是不连续的，因此它是一个数字信号。最小的数量单位就是1个。

另一类物理量在时间上和数量上的变化是连续的。这一类物理量称为模拟量，把表示模拟量的信号叫做模拟信号，工作在模拟信号下的电子电路称为模拟电路。例如，热电偶在工作时输出的电压信号就是模拟信号，因为在任何情况下被测温度都不可能发生突跳，所以测得的电压信号无论在时间上还是在数量上都是连续的。而且，这个电压信号在连续变化过程中的任何一个取值都有具体的物理意义，即表示一个相应的温度。

1.1.2　数制和码制

1. 数制

用数字量表示物理量的大小时，仅用一位数码往往不够用，因此经常要用进位计数的方法组成多位数码使用。把多位数码中每一位的构成方法以及从低位到高位的进位规则称为数制。由此可见，数制实际是计数进位制的简称。

在数字电路中经常使用的计数进制除了十进制以外，还经常使用二进制和十六进制。

(1)十进制

十进制是日常生活中和工作中最常用的计数体制。在十进制数中，每一位有0~9十个数码，所以计数的基数是10。超过9的数必须用多位数表示，其中低位和相邻高位之间的进位关系是“逢十进一”，故称为十进制。

在一个多位十进制数中，每个数码处于十进制数的不同数位时，所代表的数值是不同的。例如888的数值为8×100+8×10+8×1，其中最高位数码"8"代表数值800，中间数码"8"代表数值80，最低位数码"8"代表数值8。把100、10、1称为十进制数数位的位权值。十进制数的每个数位的位权值是10的幂。所以任何一个十进制数的数值，都可以按位权展开为：

$$(D)_{10}=k_{n-1}\times10^{n-1}+k_{n-2}\times10^{n-2}+\cdots+k_1\times10^1+k_0\times10^0+$$

$$k_{-1}\times10^{-1}+k_{-2}\times10^{-2}+\cdots+k_{-m}\times10^{-m}=\sum_{i=-m}^{n-1}k_i\times10^i \tag{1-1}$$

式中 k_i 是第 i 位的系数，可以是0～9十个数码中的任何一个。若整数部分的位数是 n，小数部分的位数为 m，则 i 包含从 $n-1$ 到0的所有正整数和从 -1 到 $-m$ 的所有负整数。例如：

$$164.75=1\times10^2+6\times10^1+4\times10^0+7\times10^{-1}+5\times10^{-2}$$

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的计数制。若以 N 取代式(1-1)中的10，即可得到 N 进制(任意进制)数展开式的普遍形式

$$(D)_N=k_{n-1}\times N^{n-1}+k_{n-2}\times N^{n-2}+\cdots+k_1\times N^1+k_0\times N^0+k_{-1}\times N^{-1}+$$

$$k_{-2}\times N^{-2}+\cdots+k_{-m}\times N^{-m}=\sum_{i=-m}^{n-1}k_i\times N^i \tag{1-2}$$

式(1-2)中 i 的取值与式(1-1)的规定相同。N 称为计数的基数，k_i 为第 i 位的系数，N^i 称为第 i 位的权。

(2)二进制

在数字电路中应用最广的是二进制。在二进制数中，每位仅有0和1两种数码，所以计数基数为2。二进制数中，低位和相邻高位间的进位是"逢二进一"，故称为二进制。在二进制数中，每个数位的位权值为2的幂。因此二进制数可以按位权展开为：

$$(D)_2=k_{n-1}\times2^{n-1}+k_{n-2}\times2^{n-2}+\cdots+k_1\times2^1+k_0\times2^0+$$

$$k_{-1}\times2^{-1}+k_{-2}\times2^{-2}+\cdots+k_{-m}\times2^{-m}$$

$$=\sum_{i=-m}^{n-1}k_i\times2^i \tag{1-3}$$

式(1-3)中，k_i 为数码0或1；n 和 m 为正整数；2^i 为 i 位的位权值。例如：

$$(101.01)_2=1\times2^2+0\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2}$$

(3)十六进制

十六进制数的每一位有十六个不同的数码，分别是0、1、2、3、4、5、6、7、8、9、A(10)、B(11)、C(12)、D(13)、E(14)、F(15)。它的计数基数是16，低位和相邻高位间的进位关系是"逢十六进一"，各数位的位权值为16的幂。因此十六进制数可以按位权展开为：

$$(D)_{16}=k_{n-1}\times16^{n-1}+k_{n-2}\times16^{n-2}+\cdots+k_1\times16^1+k_0\times16^0+$$

$$k_{-1}\times16^{-1}+k_{-2}\times16^{-2}+\cdots+k_{-m}\times16^{-m}$$

$$=\sum_{i=-m}^{n-1}k_i\times16^i \tag{1-4}$$

式(1-4)中 k_i 是第 i 位的系数，它可以是十六进制中十六个不同的数码。若整数部分的位数是 n，小数部分的位数为 m，则 i 包含从 $n-1$ 到0的所有正整数和从 -1 到 $-m$ 的所有负整数。例如：

$$(2A.7F)_{16} = 2\times16^1 + 10\times16^0 + 7\times16^{-1} + 15\times16^{-2}$$

目前在微型计算机中普遍采用 8 位、16 位和 32 位二进制并行运算，而 8 位、16 位和 32 位的二进制数可以用 2 位、4 位和 8 位的十六进制数表示，用十六进制符号书写程序十分简便。

2. 不同进制数的相互转换

(1)二—十进制转换

把二进制数转换为等值的十进制数称为二—十进制转换。转换时只要将二进制数按式(1-3)展开，然后把所有项的数值按十进制数相加，就可以得到等值的十进制数了。例如：

$$\begin{aligned}(1101.01)_2 &= 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2}\\ &= 8+4+2+1+0+0.25 = (15.25)_{10}\end{aligned}$$

上式中分别使用下脚注的 2 和 10 表示括号里的数是二进制和十进制数。有时也用 B(Binary)和 D(Decimal)代替 2 和 10 这两个脚注。

(2)十—二进制转换

把十进制数转换成等值的二进制数称为十—二进制转换。

把十进制数转换为二进制数，需把十进制数的整数部分和小数部分分别进行转换，然后再把它们合并起来。

首先讨论整数部分的转换。由十进制数整数转换为二进制数，采用逐次除以基数 2 取余数的方法。假定十进制数为$(S)_{10}$，等值的二进制数为$(k_n k_{n-1}\cdots k_0)_2$则依式(1-3)可知

$$\begin{aligned}(S)_{10} &= k_n 2^n + k_{n-1}2^{n-1} + \cdots + k_1 2^1 + k_0 2^0\\ &= 2(k_n 2^{n-1} + k_{n-1}2^{n-2} + \cdots + k_1) + k_0 \qquad (1\text{-}5)\end{aligned}$$

上式表明，若将$(S)_{10}$除以 2，则得到的商为 $k_n 2^{n-1} + k_{n-1}2^{n-2} + \cdots + k_1$，而余数即 k_0。同理，将式(1-5)中的商除以 2 得到新的商可写成

$$k_n 2^{n-1} + k_{n-1}2^{n-2} + \cdots + k_1 = 2(k_n 2^{n-2} + k_{n-1}2^{n-3} + \cdots + k_2) + k_1 \qquad (1\text{-}6)$$

由式(1-6)不难看出，若将$(S)_{10}$除以 2 所得的商再次除以 2，则所得余数即 k_1。依此类推，反复将每次得到的商再除以 2，就可求出二进制数的每一位了。

综上所述，十进制数整数转换成二进制数的步骤如下：

①将给定的十进制数整数除以 2，余数作为二进制数的最低位；

②把前一步的商再除以 2，余数作为次低位；

③重复②步骤，记下余数，直至最后商为 0，最后的除数即为二进制数的最高位。

例 1-1 将$(53)_{10}$转换成二进制数。

解：转换步骤如下：

2 | 53　　商　　余数

2 | 26　　……$1 = k_0$ 最低位

2 | 13　　……$0 = k_1$

2 | 6　　……$1 = k_2$

2 | 3　　……$0 = k_3$

2 | 1　　……$1 = k_4$

0　　……$1 = k_5$ 最高位

故$(53)_{10} = (110101)_2$。

其次讨论小数的转换。若$(S)_{10}$是一个十进制数的小数，对应的二进制小数为$(0.k_{-1}k_{-2}\cdots k_{-m})_2$，则根据式(1-3)可知

$$(S)_{10}=k_{-1}2^{-1}+k_{-2}2^{-2}+\cdots+k_{-m}2^{-m}$$

将上式两边同乘以2得到

$$2(S)_{10}=k_{-1}+(k_{-2}2^{-1}+k_{-3}2^{-2}+\cdots+k_{-m}2^{-m+1}) \tag{1-7}$$

式(1-7)说明将小数$(S)_{10}$乘以2所得乘积的整数部分即k_{-1}。

同理，将乘积的小数部分再乘以2又可以得到

$$2(k_{-2}2^{-1}+k_{-3}2^{-2}+\cdots+k_{-m}2^{-m+1})=k_{-2}+(k_{-3}2^{-1}+\cdots+k_{-m}2^{-m+1}) \tag{1-8}$$

亦即乘积的整数部分就是k_{-2}。

依此类推，将每次乘2后所得乘积的小数部分再乘以2，便可求出二进制小数的每一位。

综上所述，十进制小数转换成二进制数，采用把小数部分逐次乘以2，取乘积的整数部分作为二进制的各有关数位，乘积的小数部分继续乘以2，直至最后乘积为0或达到一定的精度为止。

例 1-2 将十进制小数$(0.375)_{10}$转换成二进制小数。

解：转换步骤如下：

$$\begin{array}{r} 0.375 \\ \times \quad\quad 2 \\ \hline 0.750 \end{array} \quad \begin{array}{l} \text{整数部分} \\ \\ \cdots\cdots 0=k_{-1} \end{array}$$

$$\begin{array}{r} 0.750 \\ \times \quad\quad 2 \\ \hline 1.500 \end{array} \quad \cdots\cdots 1=k_{-2}$$

$$\begin{array}{r} 0.500 \\ \times \quad\quad 2 \\ \hline 1.000 \end{array} \quad \cdots\cdots 1=k_{-3}$$

故$(0.375)_{10}=(0.011)_2$。

(3)二—十六进制转换

把二进制数转换成等值的十六进制数称为二—十六进制转换。

已经知道，4位二进制数构成一位十六进制数。4位二进制数恰好有16个状态，而把这4位二进制数看做一个整体时，它的进位输出又正好是逢十六进一，所以对于整数只要从低位到高位每4位二进制数分为一组，并代之以等值的十六进制数，即可得到对应的十六进制数的整数部分。而小数部分则从小数点开始从高位到低位每4位(最后不足4位补0凑足4位)二进制数分为一组，并代之以等值的十六进制数，得到对应的十六进制数的小数部分。

例 1-3 将$(01011110.10110010)_2$转换为十六进制数。

解：按照上述转换方法得到：

$$\begin{array}{cccc} (0101 & 1110 & 1011 & 0010)_2 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ (\ 5 & \mathrm{E}\ . & \mathrm{B} & 2)_{16} \end{array}$$

(4)十六—二进制转换

十六—二进制转换是指把十六进制数转换成等值的二进制数。转换时只需将十六进制数的每一位用等值的4位二进制数代替就行了。

例 1-4 将$(3AF.C5)_{16}$转换成二进制数。

解:按照上述转换方法得到:

$$
\begin{array}{ccccc}
(\,3 & A & F\,. & C & 5\,)_{16} \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
(0011 & 1010 & 1111. & 1100 & 0101)_2
\end{array}
$$

(5)十六进制数与十进制数的转换

在把十六进制数转换为十进制数时,可根据式(1-4)把各位按权展开后相加求得。在把十进制数转换为十六进制数时,可以先把十进制数转换成二进制数,然后再把得到的二进制数转换成等值的十六进制数。这两种转换方法前面已经讲过了。

3.码制

数码不仅可以表示数量的不同大小,而且还能表示不同的事物。当用数码表示不同事物时,这些数码已不再有表示数量大小的含义,而只是不同事物的代号。所以定义:用数码表示不同事物时,这些数码称为代码。

例如,在举行体育比赛时,为便于识别运动员,通常给每个运动员编一个号码。显然,这些号码仅仅表示不同的运动员,已失去了数量大小的含义。在日常生活中代码的种类还很多,如列车的编号、学号、门牌号等。

为便于记忆和处理,在编制代码时总要遵循一定的规则。编制代码时遵循的规则称为码制。

本节主要介绍二—十进制代码,简称BCD(Binary Coded Decimal)代码和格雷码。

(1)二—十进制代码(BCD代码)

由于十进制数共有0、1、2、……、9十个数码,因此至少需要4位二进制数码表示1位十进制数。4位二进制数码共有$2^4=16$种,如表1-1所示。在这十六种代码中,可以任意挑选10种表示10个十进制数码,当然方案很多,也就是有多种不同的码制。通常把这些代码统称为二—十进制代码。

表 1-1 四位二进制代码

DCBA	对应的十进制数
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

BCD码有多种,常用的几种BCD码列于表1-2中,它们的编码规则不同。

8421码是BCD码中最常用的一种。在这种编码方式中每一位二值代码的1都代表一个固定数值,把每一位的1代表的十进制数加起来,得到的结果就是它所代表的十进制数。由于代码中从左到右每一位的1分别表示8、4、2、1,所以把这种代码叫做8421码。每一位的1代表的十进制数称为这一位的位权(简称权)。8421码中每一位的位权是固定不变的,它属于恒权代码。

表 1-2　几种常见的 BCD 代码

十进制数	8421 码	余 3 码	2421 码	5211 码	余 3 循环码
0	0000	0011	0000	0000	0010
1	0001	0100	0001	0001	0110
2	0010	0101	0010	0100	0111
3	0011	0110	0011	0101	0101
4	0100	0111	0100	0111	0100
5	0101	1000	1011	1000	1100
6	0110	1001	1100	1001	1101
7	0111	1010	1101	1100	1111
8	1000	1011	1110	1101	1110
9	1001	1100	1111	1111	1010
权	8421	无权	2421	5211	无权

余 3 码的编码规则比较特殊。如果把每个余 3 码看做 4 位二进制数，则它的数值要比它所表示的十进制数多 3，故将这种代码叫做余 3 码。如果将两个余 3 码相加，所得的和将比对应表示的十进制数多 6。因此在用余 3 码做十进制加法运算时，若两个十进制数之和为 10 时，两个对应余 3 码之和正好等于二进制数的 16，于是便从高位自动产生进位信号而不必进行修正。另外从表 1-2 中还可以看出，0 和 9、1 和 8、2 和 7、3 和 6、4 和 5 的余 3 码互为反码，这对于求取对 10 的补码是很方便的。余 3 码不能由各位二进制数的权来求出其代表的十进制数，故余 3 码不是恒权代码而是无权码。

2421 码是一种恒权代码，它的 0 和 9、1 和 8、2 和 7、3 和 6、4 和 5 也互为反码。这个特点和余 3 码相仿。

5211 码是另一种恒权代码。等到学完了第 5 章中计数器的分频作用后就会发现，如果按 8421 码接成十进制计数器，则连续输入计数脉冲时，4 个触发器输出脉冲对于计数脉冲的分频比，从低位到高位依次为 5:2:1:1。可见 5211 码每一位的权正好与 8421 码十进制计数器 4 个触发器输出脉冲的分频比相对应。这种对应关系在构成某些数字系统时很有用。

余 3 循环码是一种变权码，每一位的 1 在不同代码中并不代表固定的权值。它的主要特点是相邻的两个代码之间仅有一位状态不同。因此，按余 3 循环码接成计数器时，每次状态转换过程中只有一个触发器翻转，译码时不会发生竞争冒险现象（详见组合逻辑电路部分）。

（2）格雷码

格雷码（Gray Code）的特点是：相邻两个代码之间仅有一位不同，其余各位均相同。

计数电路按格雷码计数时，每次状态的更新仅有一位代码变化，因此减少了出错的可能性。格雷码属于无权码。它有多种代码形式，其中最常用的一种是循环码。表 1-3 给出四位循环码的编码表。由表 1-3 可知，不仅相邻两个代码只有一位不同，而且首尾（0 和 15）两个代码也仅有一位不同，构成一个“循环”，故称为循环码。此外，这种代码还具有“反射性”，即以中间为对称的两个代码（例如 0 和 15、1 和 14、…7 和 8）也只有一位不同，所以又把它称为反射码。

表 1-3　四位循环码编码表

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

1.1.3　算术运算与逻辑运算

当两个二进制数码表示两个数量大小时，它们可以进行数值运算，称这种数值运算为算术运算。二进制算术运算法则和十进制算术运算法则基本相同，惟一的区别在于相邻两位之间的关系是“逢二进一”及“借一当二”，而不是十进制数的“逢十进一”及“借一当十”。

例如，两个二进制数 1001 和 0101 的算术运算有：

加法运算

$$\begin{array}{r} 1001 \\ +\ 0101 \\ \hline 1110 \end{array}$$

减法运算

$$\begin{array}{r} 1001 \\ -\ 0101 \\ \hline 0100 \end{array}$$

乘法运算

$$\begin{array}{r} 1001 \\ \times\ 0101 \\ \hline 1001 \\ 0000 \\ 1001 \\ 0000 \\ \hline 101101 \end{array}$$

除法运算

$$\begin{array}{r} 1.11001\cdots \\ 0101\,\overline{)\,1001} \\ \underline{0101} \\ 1000 \\ \underline{0101} \\ 0110 \\ \underline{0101} \\ 001000 \\ \underline{0101} \\ 011 \end{array}$$

在数字电路和数字电子计算机中，二进制数的正、负号是用 0 和 1 表示的。在定点运算情况下，以最高位作为符号位，正数为 0，负数为 1。用这种方式表示的数码称为原码。例如：

$$(\underset{\text{符号位}}{0}\ 1011001)_2 = (+59)_{10}$$

$$(\underset{\text{符号位}}{1}\ 1011001)_2 = (-59)_{10}$$

为了简化运算电路，在数字电路中两数相减的运算是用它们的补码相加完成的。二进制数的补码是这样定义的：

①最高位为符号位，正数为 0，负数为 1；

②正数的补码和它的原码相同；

③负数的补码可通过将原码的数值位逐位求反，然后在最低位上加 1 得到。

例 1-5　计算$(1001)_2-(0101)_2$。

解：根据二进制数的运算规则可知

$$
\begin{array}{r}
1001 \\
-0101 \\
\hline
0100
\end{array}
$$

在采用补码运算时，首先求出$(+1001)_2$和$(-0101)_2$的补码，它们是

$$[+1001]_{补} = \underset{符号位}{\boxed{0}}\,1001$$

$$[-0101]_{补} = \underset{符号位}{\boxed{1}}\,1011$$

然后将两个补码相加并舍去进位

$$
\begin{array}{r}
0\,1\,0\,0\,1 \\
+\ \ 1\,1\,0\,1\,1 \\
\hline
舍去\leftarrow\ 1\ \boxed{0}\,0\,1\,0\,0
\end{array}
$$

得到的结果与前面相同。这样就把减法运算转化成了加法运算。

此外，乘法运算可以用加法和移位两种操作实现，而除法运算可以用减法和移位操作实现。因此，二进制数的加、减、乘、除运算都可以用加法运算电路完成，这就大大简化了运算电路的结构。

1 位二进制数码的 0 和 1 不仅可以表示数量的大小，并可进行二进制数的算术运算，而且可以表示两种不同的逻辑状态。例如，用 0 和 1 可分别表示一件事情的是和非、真和假、好和坏，或者表示电压的高和低、脉冲信号的有和无、开关的通与断、电灯的亮与灭等等。这种只有两种对立逻辑状态的逻辑关系称为二值逻辑。

这样，0 和 1 已不再是通常的二进制数，而是代表两种逻辑状态的符号，它们的意义完全取决于事先的“约定”。例如，以 1 表示高电平，以 0 表示低电平，或相反。

当两个二进制数码 0 和 1 表示不同的逻辑状态时，它们之间可以按照某种逻辑关系进行所谓的逻辑运算。这种逻辑运算和算术运算有着本质的不同。下面将重点介绍逻辑运算的各种规律。

1.2 逻辑代数中的三种基本运算

1.2.1 逻辑代数

1849 年，英国数学家乔治·布尔(George Bool)首先提出了描述客观事物逻辑关系的数学方法——布尔代数。1938 年克劳德·香农(Claude E Shannon)将布尔代数应用到继电器开关电路的设计，因此又称为开关代数。随着数字技术的发展，布尔代数成为数字逻辑电路分析和设计的基础，又称为逻辑代数。本章所讲的逻辑代数是布尔代数在二值逻辑电路中的应用。

逻辑代数和普通代数一样也用字母表示变量，这种变量称为逻辑变量。但是，逻辑变量与普通代数中的变量有着本质的差别，逻辑变量的取值只有 0 和 1 两种可能，而且这里的 0 和 1 已不再表示数量的大小，只代表两种不同的逻辑状态。

1.2.2 逻辑代数中的三种基本逻辑运算

逻辑代数的基本运算有“与”、“或”、“非”三种。为便于理解它们的含义，首先来看一个简