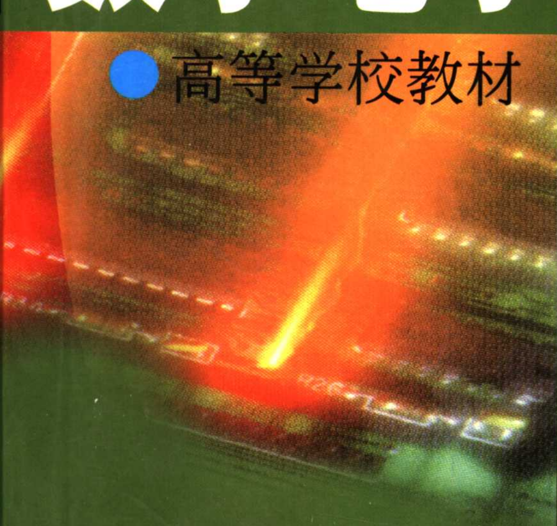
Shuzidianzijishujichu

数字电子技术基础

高等学校教材



主编 张子珍
副主编 邬祥忠

天津科学技术出版社

高等学校教材

数字电子技术基础

主　编　张子珍
副主编　邬祥忠

天津科学技术出版社

图书在版编目(CIP)数据

数字电子技术基础/张子珍主编.—天津:天津科学技术出版社,2002.1
高等学校教材
ISBN 7-5308-3124-0

Ⅰ.数... Ⅱ.张... Ⅲ.数字电路-高等学校-教材 Ⅳ.TN79

中国版本图书馆 CIP 数据核字(2001)第 058381 号

责任编辑:王定一
版式设计:雒桂芬
责任印制:张军利

天津科学技术出版社出版
出版人:王树泽
天津市张自忠路 189 号 邮编 300020 电话(022)27306314
石油管道报社印刷厂印刷
新华书店天津发行所发行
*
开本 787×1092 1/16 印张 20.25 字数 487 000
2002 年 1 月第 1 版
2002 年 1 月第 1 次印刷
定价:29.80 元

内 容 简 介

本书为适应电子信息学科、电子科学与技术学科和电气信息学科迅猛发展的形势，较好地处理了内容更新的切入点，大量精简了传统的分立元件、小规模集成电路和脉冲技术等内容，适量地增加了反映当代电子技术学科理论与技术发展前沿水平的新内容。

全书共八章，包括数字逻辑基础、集成逻辑门、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、数—模和模—数转换电路等内容。各章都配置了较多的思考题和习题，书后有附录及英汉名词术语对照。

本书可作为电子信息科学与技术类、计算机类和电气信息类自动化等专业的教科书，也可供其它相关专业和工程技术人员参考。

前　言

世界已进入信息时代。信息的处理、存储和传输,要靠电子信息技术、计算机技术和网络通信技术,而这些电子技术都是以数字技术为基础的。数字系统与数字设备已广泛应用于各个领域,因此,电子信息技术、计算机技术及电气信息技术等领域的工程技术人员,必须掌握数字系统的基础知识。

我国的经济建设正以前所未有的速度健康发展。国民经济的发展需要大量的各个层次的科技人才,因而各级各类教育已经迅猛地发展起来。教育的发展需要教材建设,编写适用的教材是教育工作者义不容辞的责任。本书就是按照教育部(原国家教育委员会)高等院校电子技术课程教学指导小组,于1993年修订的"电子技术基础课程教学基本要求"编写的,是电子类专业的技术基础课教材。

在编写过程中,力求做到基本内容符合原国家教育委员会颁发的课程教学基本要求,同时又处理好教材内容更新和基础内容相对稳定的关系;处理好教材内容的先进性与适用性的关系;既要打好基础,又要注意培养学生掌握新知识、新技术的能力。

数字电子技术是当代发展最快的学科之一。数字逻辑器件已从60年代的小规模集成电路(SSI)发展到目前的中、大规模集成电路(MSI、LSI)及超大规模集成电路(VLSI)。相应地,数字逻辑电路的设计方法也在不断地演变和发展,由原来单一的硬件逻辑设计发展成三个分支:硬件逻辑设计、软件逻辑设计和专用集成电路设计。

由于"数字电子技术基础"是一门技术性很强的专业基础课,为此在编写过程中力求做到:

1. 突出重点。重点应该是基本概念、基本知识和基本方法。逻辑代数基础、门电路组合逻辑电路、集成触发器和时序逻辑电路等是本课程的重点。逻辑代数基本定理、组合电路和时序电路的概念不仅是分析和设计数字系统的基础,也是设计大规模集成芯片的基础,尽管大规模集成电路已经成为数字系统的主体,但中、小规模集成电路仍是各类数字系统中不可缺少的。因此,作为数字技术的入门课程,本书仍以中、小规模集成电路为主的数字逻辑电路的基本理论、基本电路和基本分析、设计方法为重点。

2. 更新内容。由于专用集成电路(ASIC)是近期迅速发展起来的新逻辑器件,尤其是可编程逻辑器件(PLD)已广泛应用于数字系统设计中,这些器件的灵活性和通用性使它们已经成为研制和设计数字系统的最理想器件。因此,在编写过程中加强了这部分内容,为应用这些器件研制设计数字系统打下一定的基础。

本书内容包括数字逻辑基础、集成逻辑门、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、数—模及模—数转换电路。

书中采用国家标准图形符号，在出现图形符号的地方对其所表示的意义进行了简单明确的解释。读者在学习本教材过程中应同时学会认读常用的逻辑符号。

本书第一章由郭映、张子珍执笔，第二章由张子珍、郭映执笔，第三章由苑伯达、李艳荣执笔，第四章由严新忠、邬祥忠执笔，第五章由邬祥忠、王慧执笔，第六章由李艳荣、严新中执笔，第七章由张子珍、苑伯达执笔，第八章由王慧、邬祥忠执笔。张子珍和邬祥忠对全书进行了统稿和校订。清华大学贾培发教授对全部书稿进行了审阅，并提出了许多宝贵意见，在此谨致以诚挚的谢意。

限于编者的水平有限、经验不足，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6月

目　录

第一章　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。首先简要介绍数字电路中常用的数制与码制。然后介绍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，但就其变化规律的特点而言，不外乎两大类。

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，我们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，我们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

一、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位制除了十进制以外，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

$$(D)_{10}=k_{n-1}\times10^{n-1}+k_{n-2}\times10^{n-2}+\cdots+k_1\times10^1+k_0\times10^0+$$
$$k_{-1}\times10^{-1}+k_{-2}\times10^{-2}+\cdots+k_{-m}\times10^{-m}=\sum_{i=-m}^{n-1}k_i\times10^i \quad (1.1.1)$$

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(1.1.1)中的 10,即可得到 N 进制(任意进制)数展开式的普遍形式

$$(D)_N=k_{n-1}\times N^{n-1}+k_{n-2}\times N^{n-2}+\cdots+k_1\times N^1+k_0\times N^0+k_{-1}\times N^{-1}+$$
$$k_{-2}\times N^{-2}+\cdots+k_{-m}\times N^{-m}=\sum_{i=-m}^{n-1}k_i\times N^i \quad (1.1.2)$$

式(1.1.2)中 i 的取值与式(1.1.1)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

$$(D)_2=k_{n-1}\times2^{n-1}+k_{n-2}\times2^{n-2}+\cdots+k_1\times2^1+k_0\times2^0+$$
$$k_{-1}\times2^{-1}+k_{-2}\times2^{-2}+\cdots+k_{-m}\times2^{-m}$$
$$=\sum_{i=-m}^{n-1}k_i\times2^i \quad (1.1.3)$$

式(1.1.3)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:

$$(D)_{16}=k_{n-1}\times16^{n-1}+k_{n-2}\times16^{n-2}+\cdots+k_1\times16^1+k_0\times16^0+$$
$$k_{-1}\times16^{-1}+k_{-2}\times16^{-2}+\cdots+k_{-m}\times16^{-m}$$
$$=\sum_{i=-m}^{n-1}k_i\times16^i \quad (1.1.4)$$

式(1.1.4)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。

二、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(1.1.3)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$(k_nk_{n-1}\cdots k_0)_2$，则依式（1.1.3）可知

$$
\begin{aligned}
(S)_{10} &= k_n2^n+k_{n-1}2^{n-1}+\cdots+k_12^1+k_02^0\\
&= 2(k_n2^{n-1}+k_{n-1}2^{n-2}+\cdots+k_1)+k_0
\end{aligned}
\tag{1.1.5}
$$

上式表明，若将$(S)_{10}$除以2，则得到的商为$k_n2^{n-1}+k_{n-1}2^{n-2}+\cdots+k_1$，而余数即$k_0$。同理，将式（1.1.5）中的商除以2得到新的商可写成

$$
k_n2^{n-1}+k_{n-1}2^{n-2}+\cdots+k_1 = 2(k_n2^{n-2}+k_{n-1}2^{n-3}+\cdots+k_2)+k_1 \tag{1.1.6}
$$

由式（1.1.6）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 1.1.1】 将$(53)_{10}$转换成二进制数。

解 转换步骤如下：

		余数
2	53	
2	26 ← 商	$\cdots\cdots 1=k_0$ 最低位
2	13	$\cdots\cdots 0=k_1$
2	6	$\cdots\cdots 1=k_2$
2	3	$\cdots\cdots 0=k_3$
2	1	$\cdots\cdots 1=k_4$
	0	$\cdots\cdots 1=k_5$ 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（1.1.3）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

$$2(S)_{10}=k_{-1}+(k_{-2}2^{-1}+k_{-3}2^{-2}+\cdots+k_{-m}2^{-m+1}) \tag{1.1.7}$$

式（1.1.7）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

$$2(k_{-2}2^{-1}+k_{-3}2^{-2}+\cdots+k_{-m}2^{-m+1})=k_{-2}+(k_{-3}2^{-1}+\cdots+k_{-m}2^{-m+1}) \tag{1.1.8}$$

亦即乘积的整数部分就是 k_{-2}。

依次类推，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 1.1.2】 将十进制小数$(0.375)_{10}$转换成二进制小数。

解 转换步骤如下：

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

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

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

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 1.1.3】 将$(01011110.10110010)_2$转换为十六进制数。

解 按照上述转换方法得到

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

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 1.1.4】 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—十进制转换

在把十六进制数转换为十进制数时,可根据式(1.1.4)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。

三、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 1.1.1 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 1.1.1 四位二进制代码

DCDA	对应的十进制数
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	8
1 0 0 1	9
1 0 1 0	10
1 0 1 1	11
1 1 0 0	12
1 1 0 1	13
1 1 1 0	14
1 1 1 1	15

BCD 码有多种,常用的几种 BCD 码列于表 1.1.2 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 1.1.2 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 1.1.2 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数值。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 1.1.3 给出四位循环码的编码表。由表 1.1.3 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 1.1.3　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：

$$(\boxed{0}\,1011001)_2=(+59)_{10}$$

符号位

$$(\boxed{1}\,1011001)_2=(-59)_{10}$$

符号位

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 1.1.5】 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

$$\begin{array}{rr} & 01001 \\ & +\ 11011 \\ \hline 舍去\leftarrow & 1\,\boxed{0}0100 \end{array}$$

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。这种只有两种对应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