

21世纪高等职业技术教育规划教材

——电子信息类

数字电子技术

SHUZI
DIANZI JISHU

主 编 周 华
副主编 李奇琳 李 强



西南交通大学出版社
[Http://press.swjtu.edu.cn](http://press.swjtu.edu.cn)

21 世纪高等职业技术教育规划教材——电子信息类

数字电子技术

主 编 周 华

副主编 李奇琳 李 强

主 审 王永奇

西南交通大学出版社

· 成 都 ·

内 容 简 介

本书共 7 章, 包括数字电子技术基础、组合逻辑电路、时序逻辑电路、脉冲信号的产生与变换、数/模和模/数转换、大规模集成电路、数字电路综合应用。

本书注重“讲、学、做”统一协调, 将理论知识与应用及技能训练有机结合, 叙述简练清楚, 实例与知识点结合恰当, 例题分析透彻, 习题安排合理, 实验与课程设计内容都是精心设计的。

本书可供高等职业院校、高等专科学校、成人高等学校以及本科院校电气类、电工类、通信类、自动控制类、计算机类专业使用, 也可供其他专业及参加自学考试的同学和有关技术人员参考。

图书在版编目 (C I P) 数据

数字电子技术 / 周华主编. —成都: 西南交通大学出版社, 2009.7

21 世纪高等职业技术教育规划教材. 电子信息类
ISBN 978-7-5643-0310-5

I. 数… II. 周… III. 数字电路—电子技术—高等学校: 技术学校—教材 IV. TN79

中国版本图书馆 CIP 数据核字 (2009) 第 117737 号

21 世纪高等职业技术教育规划教材——电子信息类

数字电子技术

主编 周 华

*

责任编辑 黄淑文

封面设计 墨创文化

西南交通大学出版社出版发行

(成都市二环路北一段 111 号 邮政编码: 610031 发行部电话: 028-87600564)

<http://press.swjtu.edu.cn>

成都蜀通印务有限责任公司印刷

*

成品尺寸: 185 mm×260 mm 印张: 14

字数: 350 千字 印数: 1—3 000 册

2009 年 7 月第 1 版 2009 年 7 月第 1 次印刷

ISBN 978-7-5643-0310-5

定价: 24.00 元

图书如有印装质量问题 本社负责退换
版权所有 盗版必究 举报电话: 028-87600562

前言

本教材是《21 世纪高等职业技术教育规划教材——电子信息类》规划教材之一。

高等职业教育专业教学改革的核心是课程体系和教学内容的改革,也是专业教学改革的重点和难点。教材改革在这方面下了很多工夫,我们在认真总结、吸取自身多年教学经验和国内外职业教育课程改革经验的基础上,在教学内容、教学模式、教材体系结构及教学方法、手段等方面具有较大创新,并能较好地体现现代职业教育教学观念。

在教材内容的编写上坚持“以应用为目的,以必需、够用为度”和“少而精”的原则。在编写过程中采用理论与实际相结合的方法,充分体现能力本位的思想,注重在学习理论知识的基础上重点培养学生的实践能力,既要教会学生运用理论去观察和分析问题,又要培养学生解决问题的能力。

本书内容的选用和编排有以下几大特点:

1. 本书全面而系统地介绍了数字电路的基本概念、基本理论和基本分析方法。阐述准确简明、重点突出,脉络清晰。删去复杂的理论分析,注重培养分析问题、解决问题的能力。
2. 本书注重理论联系工程实际。内容讲解、例题选用尽量贴近实际应用,有数字电路实验及课程设计,以培养技能型、应用型人才为目标,强化学生工程技术的应用能力。
3. 本书体现时代特征,删去老化的知识点,突出高职教育的特点和培养目标,尽量多介绍电子技术领域的有关新知识和技术,使学生能学到新颖的、实用的知识,有利于培养学生的创新精神。
4. 本书内容通俗易懂、层次清晰,文字深入浅出,版面新颖、活泼,对学生有吸引力。
5. 本书配有电子教案,有利于教师教学和学生自学。

本书主要是介绍数字电路的基本概念、基本公式及工程应用的一门技术基础课。全书共分为 7 章,内容包括:数字电子技术基础、组合逻辑电路、时序逻辑电路、脉冲信号的产生与变换、A/D 和 D/A 转换、大规模集成电路、数字电路综合应用。本书可供电气类、电工类、通信类等各专业人员及非电专业人员使用。

教材编写参与教师有贵州电子信息职业技术学院周华、王永奇和陆忠梅,成都农业科技职业学院李奇琳,贵州航天职业技术学院李强、杨代民。

第 1 章、第 5 章由周华编写;第 2 章由李强编写;第 3 章、第 7 章由李奇琳编写;第 4 章由陆忠梅编写;第 6 章由杨代民编写。

本书由周华担任主编,李奇琳、李强任副主编;王永奇任主审;周华负责全书统稿。

由于编者水平有限,错误和不妥之处在所难免,恳请读者批评指正。

编者

2009 年 5 月

目 录

第 1 章 数字电子技术基础	1
1.1 数字电路概述	1
1.2 数制与编码	3
1.3 逻辑代数基础	8
1.4 逻辑函数	16
1.5 逻辑函数的化简	20
1.6 逻辑函数的表示方法之间的转换	29
1.7 分立元件门电路	32
1.8 TTL 集成门电路	40
1.9 CMOS 集成门电路	48
小 结	56
习 题	57
第 2 章 组合逻辑电路	62
2.1 组合逻辑电路的分析与设计方法	62
2.2 加法器	70
2.3 数值比较器	73
2.4 编码器	76
2.5 译码器	82
2.6 数据选择器	90
2.7 数据分配器	96
小 结	98
习 题	99
第 3 章 时序逻辑电路	102
3.1 触发器	102
3.2 时序逻辑电路的分析和设计方法	112
3.3 计数器	117
3.4 寄存器	124
3.5 顺序脉冲发生器	127
小 结	129
习 题	130

第 4 章 脉冲信号的产生与变换	133
4.1 集成 555 定时器	133
4.2 单稳态触发器	135
4.3 施密特触发器	140
4.4 多谐振荡器	143
小 结	146
习 题	147
第 5 章 数/模和模/数转换	148
5.1 概 述	148
5.2 D/A 转换器 (DAC)	148
5.3 A/D 转换器 (ADC)	153
小 结	162
习 题	163
第 6 章 大规模集成电路	164
6.1 只读存储器 (ROM)	164
6.2 可编程逻辑器件 (PLD)	174
6.3 随机存取存储器 (RAM)	179
小 结	181
习 题	181
第 7 章 数字电路综合应用	182
7.1 常用仪器与设备的使用方法	182
7.2 电子电路设计的基础知识	185
7.3 数字电路实训	192
7.4 数字电子技术课程设计	202
参考文献	218

数字电子技术基础

提 要 在数字电路中,进行逻辑分析与设计的数学工具是逻辑代数。本章介绍数字电路的特点、数制与编码、逻辑代数及其运算、逻辑代数的公式和运算规则、逻辑函数表示方法及相互转换、逻辑函数的公式化简法、逻辑函数的卡诺图化简法、门电路。

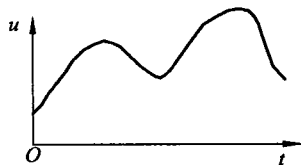
1.1 数字电路概述

1.1.1 数字信号与数字电路

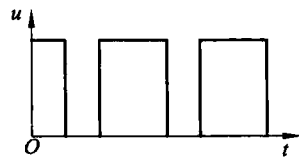
1. 数字信号与数字电路

电子电路中的电信号可以分为两类。一类是在连续的时间范围内有定义且幅度连续变化的信号,这种连续变化的信号称为模拟信号,例如,时间、压强、路程、温度等物理量通过传感器变成的电信号,模拟语音的音频信号和模拟图像的视频信号等。对模拟信号进行传输、处理的电子线路称为模拟电路,如放大器、信号发生器等。另一类是时间和幅度都是离散的、不连续的信号,称为数字信号。对数字信号进行传输、处理的电子线路称为数字电路,如数字万用表、数字电子钟等都是由数字电路组成的。图 1.1.1 所示是模拟信号和数字信号的波形图。

数字电路被广泛应用于数字电子计算机、数字通信系统、数字式仪表等领域。数字电路大致包括信号的产生、放大、整形、传送、控制、存储、计数、运算等组成部分。



(a) 模拟信号的波形



(b) 数字信号的波形

图 1.1.1 模拟信号和数字信号的波形

2. 数字电路的优点

与模拟电路相比,数字电路具有以下显著的优点:

- ① 结构简单,便于集成化、系列化生产,成本低廉,使用方便。

- ② 抗干扰性强, 可靠性高, 精度高。
- ③ 处理功能强, 不仅能实现数值运算, 还可以实现逻辑运算和判断。
- ④ 可编程数字电路可以很容易地实现各种算法, 具有很大的灵活性。
- ⑤ 数字信号更易于存储、加密、压缩、传输和再现。

1.1.2 数字电路的特点与分类

1. 数字电路的特点

由于数字电路的工作信号是不连续的数字信号, 反映在电路上只有高电位和低电位两种状态, 所以在数字电路中二极管和三极管一般都是工作在开关状态。开关的接通与断开两种状态, 用二极管和三极管的导通与截止来实现。在数字电路中, 通常将高电位称为高电平, 低电位称为低电平。在实际数字电路中, 高电平通常为 $+3.5\text{ V}$ 左右, 低电平通常为 $+0.3\text{ V}$ 左右。为了分析的方便, 而且由于数字电路采用二进制数来进行信息的传输和处理, 因此在数字电路中分别用“1”和“0”来表示高电平和低电平。这种高电平对应“1”、低电平对应“0”的关系称为正逻辑关系。也可用高电平对应“0”, 低电平对应“1”, 这种关系称为负逻辑关系。本书中所采用的都是正逻辑关系。

数字电路研究的主要问题是输出信号的状态(0 或 1)与输入信号的状态(0 或 1)之间的关系。这种关系是一种因果关系, 也就是所谓的逻辑关系, 即电路的逻辑功能。所以, 在数字电路中不能采用模拟电路的分析方法, 而是以逻辑代数作为主要工具, 利用真值表、逻辑表达式、波形图等来表示电路的逻辑功能, 所以数字电路又称为逻辑电路。

2. 数字电路的分类

数字集成电路是将电路所有的器件和连接线制作在一块半导体基片(芯片)上而成。通常以“门”为最小单位, 按“集成度”将数字集成电路分成:

小规模集成电路 SSI: 一块芯片上含 $1\sim 100$ 个门。

中规模集成电路 MSI: 一块芯片上含 $100\sim 1\,000$ 个门。

大规模集成电路 LSI: 一块芯片上含 $1\,000\sim 10\,000$ 个门。

超大规模集成电路 VLSI: 一块芯片上含 $10^4\sim 10^6$ 个门。

如果集成逻辑门电路是以双极型晶体管(电子和空穴两种载流子均参与导电)为基础制成的, 则称为双极型集成逻辑门电路。它主要有以下几种类型: 晶体管-晶体管逻辑门 TTL; 高阈值逻辑门 HTL; 射极耦合逻辑门 ECL。

如果集成逻辑门电路是以单极型晶体管(只有一种极性的载流子参与导电: 电子或空穴)为基础制成的, 则称为单极型集成逻辑门电路。目前应用最为广泛的是金属-氧化物-半导体场效应管逻辑电路, 简称 MOS 集成电路, 可分为 PMOS、NMOS 和 CMOS 集成电路。

1.2 数制与编码

1.2.1 进位计数制

进位计数制也叫位置计数制，简称数制。其计数方法是把数划分为不同的数位，当某一数位累计到一定数量之后，该位又从零开始，同时向高位进位。在这种计数制中，同一个数码在不同的数位上所表示的数值是不同的。进位计数制可以用少量的数码表示较大的数，因而被广泛采用。

进位基数：在一个数位上，规定使用的数码符号的个数叫该进位计数制的进位基数或进位模数，记作 R 。例如十进制，每个数位规定使用的数码符号为 0, 1, 2, ..., 9, 共 10 个，故其进位基数 $R=10$ 。

数位的权值：某个数位上数码为 1 时所表征的数值，称为该数位的权值，简称“权”。各个数位的权值均可表示成 R^i 的形式，其中 R 是进位基数， i 是各数位的序号。序号按如下方法确定：整数部分，以小数点为起点，自右向左依次为 0, 1, 2, ..., $n-1$ ；小数部分，以小数点为起点，自左向右依次为 $-1, -2, \dots, -m$ 。 n 是整数部分的位数， m 是小数部分的位数。

某个数位上的数码 a_i 所表示的数值等于数码 a_i 与该位的权值 R^i 的乘积。所以， R 进制的数 N 可写为：

$$(N)_R = a_{n-1}a_{n-2}\cdots a_2a_1a_0.a_{-1}a_{-2}\cdots a_{-m}$$

又可以写成如下多项式的形式：

$$\begin{aligned}(N)_R &= a_{n-1}R^{n-1} + a_{n-2}R^{n-2} + \cdots + a_2R^2 + a_1R^1 + a_0R^0 + a_{-1}R^{-1} + a_{-2}R^{-2} + \cdots + a_{-m}R^{-m} \\ &= \sum_{i=-m}^{n-1} a_i R^i\end{aligned}$$

1.2.2 常用数制

1. 十进制

在十进制中，每个数位规定使用的数码为 0, 1, 2, ..., 9, 共 10 个，故其进位基数 R 为 10。其计数规则是“逢十进一”。各位的权值为 10^i ， i 是各数位的序号。

十进制数用下标“D”表示，也可省略。例如：

$$(478.298)_D = 4 \times 10^2 + 7 \times 10^1 + 8 \times 10^0 + 2 \times 10^{-1} + 9 \times 10^{-2} + 8 \times 10^{-3}$$

2. 二进制

在二进制中，每个数位规定使用的数码为 0, 1, 共 2 个数码，故其进位基数 R 为 2。其计数规则是“逢二进一”。各位的权值为 2^i ， i 是各数位的序号。

二进制数用下标“B”表示。例如：

$$(1001.01)_B = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

二进制数只有 0 和 1 两个数码，机器实现容易，并且算术运算也很简单，因而二进制是数字系统唯一认识的代码，在数字电路中获得广泛应用。但用二进制表示一个数时，位数多，读写不方便。

3. 八进制

在八进制中，每个数位上规定使用的数码为 0, 1, 2, 3, 4, 5, 6, 7，共 8 个，故其进位基数 R 为 8。其计数规则为“逢八进一”。各位的权值为 8^i ， i 是各数位的序号。

八进制数用下标“O”表示。例如：

$$(732.62)_O = 7 \times 8^2 + 3 \times 8^1 + 2 \times 8^0 + 6 \times 8^{-1} + 2 \times 8^{-2}$$

因为 $2^3=8$ ，因而三位二进制数可用一位八进制数表示。

4. 十六进制

在十六进制中，每个数位上规定使用的数码符号为 0, 1, 2, ..., 9, A, B, C, D, E, F，共 16 个，故其进位基数 R 为 16。其计数规则是“逢十六进一”。各位的权值为 16^i ， i 是各个数位的序号。

十六进制数用下标“H”表示，例如：

$$\begin{aligned} (A3D.3C)_H &= A \times 16^2 + 3 \times 16^1 + D \times 16^0 + 3 \times 16^{-1} + C \times 16^{-2} \\ &= 10 \times 16^2 + 3 \times 16^1 + 13 \times 16^0 + 3 \times 16^{-1} + 12 \times 16^{-2} \end{aligned}$$

因为 $2^4=16$ ，所以四位二进制数可用一位十六进制数表示。

表 1.2.1 示出了几种数制之间的对应关系。

表 1.2.1 几种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
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

1.2.3 数制之间的转换

1. 二进制数与十进制数之间的转换

二进制数转换成十进制数时,采用按权展开法。只要将二进制数按权展开,然后将各项数值按十进制数相加,便可得到等值的十进制数。

例 1.2.1 将二进制数 $(10110.11)_2$ 转换成十进制数。

解 $(10110.11)_2 = 1 \times 2^4 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^{-1} + 1 \times 2^{-2} = (22.75)_{10}$

同理,若将任意进制数转换为十进制数,只需将数 $(N)_R$ 写成按权展开的多项式表示式,并按十进制规则进行运算,便可求得相应的十进制数 $(N)_{10}$ 。

2. 十进制数转换为其他进制数

1) 整数转换

整数转换,采用基数连除法。把十进制整数 N 转换成 R 进制数的步骤如下:

- ① 将 N 除以 R ,记下所得的商和余数。
- ② 将上一步所得的商再除以 R ,记下所得商和余数。
- ③ 重复做第②步,直到商为 0。
- ④ 将各个余数转换成 R 进制的数码,并按照与运算过程相反的顺序把各个余数排列起来,即为 R 进制的数。

例 1.2.2 将十进制数 $(57)_{10}$ 转换成二进制数。

解

$$\begin{array}{rcl} 2 & \overline{) 57} & \text{余数} \\ 2 & \overline{) 28} & \cdots 1 = a_0 \\ 2 & \overline{) 14} & \cdots 0 = a_1 \\ 2 & \overline{) 7} & \cdots 0 = a_2 \\ 2 & \overline{) 3} & \cdots 1 = a_3 \\ 2 & \overline{) 1} & \cdots 1 = a_4 \\ & 0 & \cdots 1 = a_5 \end{array}$$

$$(57)_{10} = (111001)_2$$

2) 纯小数转换

纯小数转换,采用基数连乘法。把十进制的纯小数 M 转换成 R 进制数的步骤如下:

- ① 将 M 乘以 R ,记下整数部分。
- ② 将上一步乘积中的小数部分再乘以 R ,记下整数部分。
- ③ 重复做第②步,直到小数部分为 0 或者满足精度要求为止。
- ④ 将各步求得的整数转换成 R 进制的数码,并按照与运算过程相同的顺序排列起来,即为所求的 R 进制数。

例 1.2.3 将十进制数小数 $(0.724)_{10}$ 转换成二进制小数。

解

$$\begin{array}{r}
 0.724 \\
 \times \quad 2 \quad \text{整数} \\
 \hline
 1.448 \cdots \cdots 1 = a_{-1} \\
 0.448 \\
 \times \quad 2 \\
 \hline
 0.896 \cdots \cdots 0 = a_{-2} \\
 \times \quad 2 \\
 \hline
 1.792 \cdots \cdots 1 = a_{-3} \\
 0.792 \\
 \times \quad 2 \\
 \hline
 1.584 \cdots \cdots 1 = a_{-4} \\
 \\
 (0.724)_{10} = (0.1011)_2
 \end{array}$$

可见，小数部分乘 2 取整的过程，不一定能使最后乘积为 0，因此转换值存在误差。通常在二进制小数的精度已达到预定的要求时，运算便可结束。

将一个带有整数和小数的十进制数转换成二进制数时，必须将整数部分和小数部分分别按除 2 取余法和乘 2 取整法进行转换，然后再将两者的转换结果合并起来即可。

同理，若将十进制数转换成任意 R 进制数 $(N)_R$ ，则整数部分转换采用除 R 取余法，小数部分转换采用乘 R 取整法。

3. 二进制数与八进制数、十六进制数之间的相互转换

八进制数和十六进制数的基数分别为 $8=2^3$ ， $16=2^4$ ，所以三位二进制数相当于一位八进制数，四位二进制数相当于一位十六进制数，它们之间的相互转换是很方便的。

二进制数转换成八进制数的方法是从小数点开始，分别向左、向右，将二进制数按每三位一组分组（不足三位的补 0），然后写出每一组等值的八进制数。

例 1.2.4 将二进制数 $(111001.011101)_B$ 转换成八进制数。

解 $(111\ 001.011\ 101)_B = (71.35)_O$

二进制数转换成十六进制数的方法是从小数点开始，分别向左、向右，将二进制数按每四位一组分组（不足四位的补 0），然后写出每一组等值的十六进制数。

例 1.2.5 将二进制数 $(1001.10110101011)_B$ 转换成十六进制数。

解 $(1001.1011\ 0101\ 011)_B = (9.B56)_H$

八进制数、十六进制数转换为二进制数的方法可以采用与前面相反的步骤，即只要按原来顺序将每一位八进制数（或十六进制数）用相应的三位（或四位）二进制数代替即可。

例 1.2.6 将八进制数 $(115.753)_O$ 转换成二进制数。

解 $(115.753)_O = (001\ 001\ 101.111\ 101\ 011)_B$

例 1.2.7 将十六进制数 $(A6D.8F)_H$ 转换成二进制数。

解 $(A6D.8F)_H = (1010\ 0110\ 1101.1000\ 1111)_B$

1.2.4 编 码

在数字系统中，需要把十进制的数值、不同的文字、符号等其他信息用二进制数码来表示才能处理。用来表示其他信息的二进制数码称为代码。建立这种代码与其他信息一一对应的关系称为编码。

1. 二-十进制码（BCD 码）

二-十进制码是用二进制码元来表示十进制数符“0~9”的代码，简称 BCD 码（Binary Coded Decimal 的缩写）。

用二进制码元来表示“0~9”这 10 个数符，必须用四位二进制码元来表示，而四位二进制码元共有 16 种组合，从中取出 10 种组合来表示“0~9”的编码方案有很多种。几种常用的 BCD 码如表 1.2.2 所示。若某种代码的每一位都有固定的“权值”，则称这种代码为有权代码；否则，叫无权代码。

表 1.2.2 几种常见的 BCD 码

十进制数	8421 码	5421 码	2421 码	余 3 码	BCD Gray 码
0	0000	0000	0000	0011	0000
1	0001	0001	0001	0100	0001
2	0010	0010	0010	0101	0011
3	0011	0011	0011	0110	0010
4	0100	0100	0100	0111	0110
5	0101	1000	1011	1000	0111
6	0110	1001	1100	1001	0101
7	0111	1010	1101	1010	0100
8	1000	1011	1110	1011	1100
9	1001	1100	1111	1100	1000

1) 8421BCD 码

8421BCD 码是有权码，各位的权值分别为 8，4，2，1。虽然 8421BCD 码的权值与四位自然二进制码的权值相同，但二者是两种不同的代码。8421BCD 码只是取用了四位自然二进制代码的前 10 种组合。

例 1.2.8 将十进制数 $(845.16)_D$ 转换成 8421BCD 码。

解 $(845.16)_D = (1000\ 0100\ 0101.0001\ 0110)_{8421BCD}$

2) 余 3 码

余 3 码是 8421BCD 码的每个码组加 0011 形成的。其中的 0 和 9，1 和 8，2 和 7，3 和 6，4 和 5，各对码组相加均为 1111，具有这种特性的代码称为自补代码。余 3 码各位无固定权值，故属于无权码。

3) 2421 码

2421BCD 码的各位权值分别为 2, 4, 2, 1, 故 2421 码是有权码, 也是一种自补代码。

用 BCD 码表示十进制数时, 只要把十进制数的每一位数码, 分别用 BCD 码取代即可。反之, 若要知道 BCD 码代表的十进制数, 只要把 BCD 码以小数点为起点向左、向右每四位分一组, 再写出每一组代码代表的十进制数, 并保持原排序即可。

4) 格雷 (Gray) 码

具有如下特点的代码叫格雷码: 任何相邻的两个码组 (包括首、尾两个码组) 中, 只有一个码元不同。

在编码技术中, 把两个码组中不同的码元的个数叫做这两个码组的距离, 简称码距。由于格雷码任意相邻两个码组的距离均为 1, 故又称之为单位距离码。另外, 由于首尾两个码组也具有单位距离特性, 因而格雷码也叫循环码。格雷码属于无权码。

格雷码的单位距离特性可以降低其产生错误的概率, 并且能提高其运行速度。例如, 为完成十进制数 7 加 1 的运算, 当采用四位自然二进制码时, 计数器应由 0111 变为 1000, 由于计数器中各元件特性不可能完全相同, 因而各位数码不可能同时发生变化, 可能会瞬间出现过性的错码。变化过程可能为 0111→1111→1011→1001→1000。虽然最终结果是正确的, 但在运算过程中出现了错码 1111, 1011, 1001, 这会造成数字系统的逻辑错误, 而且使运算速度降低。若采用格雷码, 由 7 变成 8, 只有一位发生变化, 就不会出现上述错码, 而且运算速度会明显提高。

表 1.2.1 中给出的格雷码, 具有反射特性, 即按表中所示的对称轴, 除最高位互补反射外, 其余低位码元以对称轴镜像反射。利用这一特性, 可以方便地构成位数不同的格雷码。

1.3 逻辑代数基础

在客观世界中, 事物的发展变化通常都是有一定因果关系的。电灯的亮与灭取决于电源是否接通, 如果电源接通了, 电灯就会亮, 否则就灭。这里电源接通与否是因, 电灯亮与不亮是果。这种因果关系, 一般称为逻辑关系。

逻辑代数又称布尔代数或开关代数, 是英国数学家乔治·布尔在 1847 年首先创立的。逻辑代数是研究逻辑函数与逻辑变量之间规律的一门应用数学, 是分析和设计数学逻辑电路的数学工具。

因为数字电路的输出信号与输入信号之间的关系就是逻辑关系, 所以数字电路的工作状态可以用逻辑关系来描述。逻辑代数和普通代数不同, 逻辑代数中, 有些运算规则与普通代数相同, 有些则完全不同。在学习过程中, 要加以注意。

1.3.1 逻辑代数中的逻辑变量

由于事物包含相互对立而又互相联系的两个方面, 如导通与断开、好与坏、亮与暗等, 所以在逻辑代数中, 为了描述事物两种对立的逻辑状态, 采用的是仅有两个取值的变量, 这种变量称为逻辑变量。

逻辑变量和普通变量一样，都是用字母表示。但是，它又和普通代数变量有本质的区别：我们所研究的逻辑变量的取值只有 0 和 1 两种可能，而且这里的 0 和 1 不是表示数值的大小，而是表示逻辑变量的两种状态。

可见，在数字电路中的二进制数码，有时可作二进制数表示数值的大小，此时它们之间可以进行算术运算；有时还可以作为逻辑变量表示不同的逻辑状态，此时它们之间只能按照某种逻辑关系进行逻辑运算。

下面我们将介绍三种基本的逻辑运算。

1.3.2 逻辑代数中的基本逻辑运算

逻辑代数有三种基本运算：“与”运算，“或”运算和“非”运算。

1. 与运算

只有当决定事物结果的所有条件全部具备时，结果才会发生，这种逻辑关系称为与逻辑关系（又称为与运算）。与逻辑模型电路如图 1.3.1 所示， A 、 B 是两个串联开关， Y 是灯，用开关控制灯亮和灭的关系如表 1.3.1 所示。从表 1.3.1 中可知，只有当两个开关全都接通时，灯才会亮。因此它们满足与逻辑关系。

如果用二值量中的 1 来表示灯亮和开关闭合，用 0 表示灯灭和开关断开，则可得到如表 1.3.2 所示的与逻辑真值表。

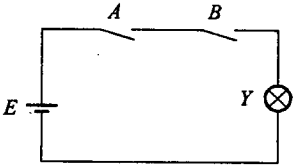


图 1.3.1 与逻辑电路图

表 1.3.1 与逻辑关系表

A	B	Y
断	断	灭
断	通	灭
通	断	灭
通	通	亮

表 1.3.2 与逻辑真值表

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

表 1.3.2 是将输入变量各种取值的组合和相应的函数值排列而成的表格。这种反映开关通断状态和电灯亮灭之间逻辑关系的表格称为逻辑真值表，简称真值表。

与运算又称“逻辑乘”。与运算的逻辑表达式为：

$$Y=A \cdot B$$

式中，“ $\cdot$ ”表示 A 、 B 的与运算，通常与运算符“ $\cdot$ ”可以省略，写为 $Y=AB$ 。上式读作“ Y 等于 A 与 B ”，或者“ Y 等于 A 乘 B ”。

由与运算的逻辑表达式 $Y=A \cdot B$ 或表 1.3.2 所示的真值表，可知与逻辑的运算规律为：

$$0 \cdot 0=0, 0 \cdot 1=0, 1 \cdot 0=0, 1 \cdot 1=1$$

即只有当 A 和 B 都是 1 时， Y 才为 1；否则 Y 为 0。总结为输入有 0 得 0，全 1 得 1。

与运算的逻辑符号如图 1.3.2 所示。

与逻辑的波形图如图 1.3.3 所示，该图直观地描述了任意时刻输入与输出之间的对应关系及变化的情况。

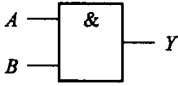


图 1.3.2 与逻辑符号图

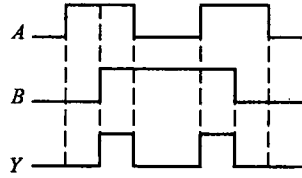


图 1.3.3 与逻辑波形图

2. 或运算

当决定事物结果的几个条件中，只要有一个或一个以上条件得到满足，结果就会发生，这种逻辑关系称为或逻辑（又称为或运算）。或逻辑模型电路如图 1.3.4 所示。图中， A 、 B 是两个并联开关， Y 是灯。用开关控制灯亮和灭的关系如表 1.3.3 所示。从表中可知，只要两个开关有一个接通，灯就会亮，因此满足或逻辑关系。

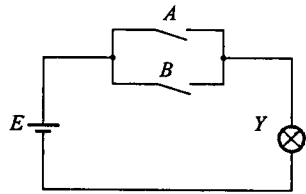


图 1.3.4 或逻辑电路图

如果用 1 表示灯亮和开关闭合，用 0 表示灯灭和开关断开，则可得到或逻辑关系真值表如表 1.3.4 所示。

表 1.3.3 或逻辑关系表

A	B	Y
断	断	灭
断	通	亮
通	断	亮
通	通	亮

表 1.3.4 或逻辑真值表

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

或运算又称“逻辑加”。或运算的逻辑表达式为：

$$Y = A + B$$

式中，“+”表示 A 、 B 的或运算。上式读作“ Y 等于 A 或 B ”，或者“ Y 等于 A 加 B ”。

由或运算的逻辑表达式 $Y = A + B$ 或表 1.3.4 所示的真值表，可知或逻辑的运算规律为：

$$0 + 0 = 0, \quad 0 + 1 = 1, \quad 1 + 0 = 1, \quad 1 + 1 = 1$$

即只有当 A 和 B 都是 0 时， Y 才为 0；否则 Y 为 1。总结为输入有 1 得 1，全 0 得 0。

或运算的逻辑符号如图 1.3.5 所示。

或逻辑的波形图如图 1.3.6 所示。该图直观地描述了任意时刻输入与输出之间的对应关系及变化的情况。

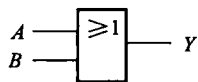


图 1.3.5 或逻辑符号

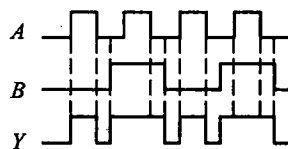


图 1.3.6 双输入端或门波形图

3. 非运算

在事件中，结果总是和条件呈相反状态，这种逻辑关系称为非逻辑（又称为非运算）。非逻辑模型电路如图 1.3.7 所示。图 1.3.7 中， A 是开关， Y 是灯。用开关控制灯亮和灭的关系如表 1.3.5 所示。从表中可知，如果开关 A 闭合，灯就灭，开关 A 断开，灯就亮，因此满足非逻辑关系。

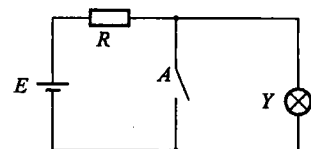


图 1.3.7 非逻辑电路图

如果用 1 表示灯亮和开关闭合，用 0 表示灯灭和开关断开，则可得到非逻辑关系真值表如表 1.3.6 所示。

表 1.3.5 非逻辑的关系表

A	Y
断	亮
通	灭

表 1.3.6 非逻辑的真值表

A	Y
0	1
1	0

非运算又称“反运算”。非运算的逻辑表达式为：

$$Y = \bar{A}$$

式中，“ $\bar{}$ ”表示 A 的非运算或者反运算。上式读作“ Y 等于 A 非”，或者“ Y 等于 A 反”。

显然非逻辑的运算规律为：

$$\bar{0} = 1, \quad \bar{1} = 0$$

即 0 变 1、1 变 0，始终相反。

非运算的逻辑符号如图 1.3.8 所示。

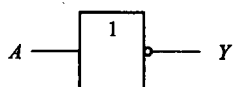


图 1.3.8 非逻辑符号

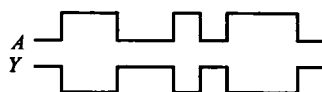


图 1.3.9 非逻辑波形图

1.3.3 逻辑代数中的五种复合逻辑运算

与、或、非三种逻辑运算是最基本的逻辑函数。可是实际的逻辑问题要比它们复杂得多。因此，在数字逻辑电路中，还常直接使用一些复合逻辑函数及其逻辑符号。