国家一级出版社
全国百佳图书出版单位
China University of Mining and Technology Press

责任编辑：齐　畅
封面设计：黄燕美

数字电子技术

ISBN 978-7-5646-7140-2

定价：59.90元

高等院校电子信息系列精品教材

数字电子技术

主编　马明涛　张　杰

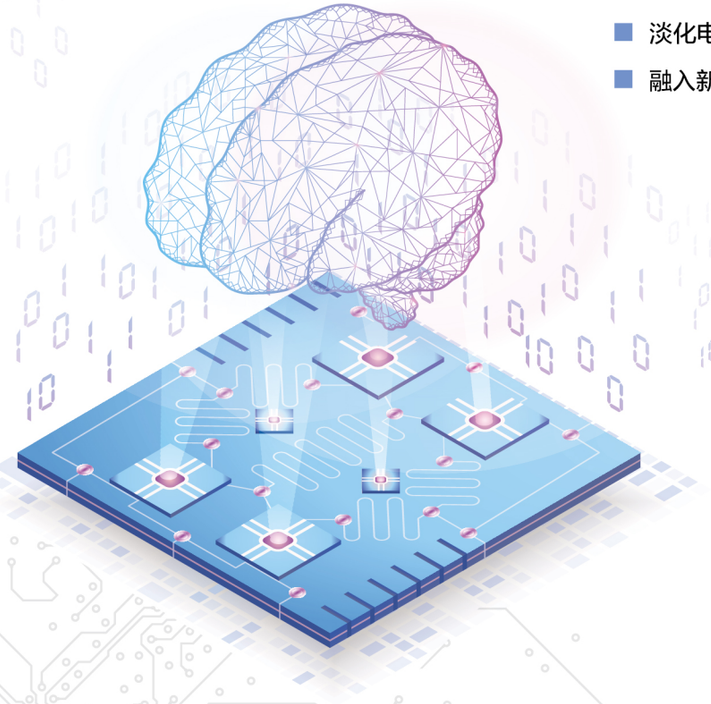
中国矿业大学出版社
China University of Mining and Technology Press

高等院校电子信息系列精品教材

数字电子技术

主编　马明涛　张　杰

- 优化内容编排，聚焦集成电路设计
- 立足数电基础课程，兼顾教学与实践
- 淡化电路工作原理，强调外特性分析
- 融入新概念、新器件、新技术、新方法



中国矿业大学出版社
China University of Mining and Technology Press

高等院校电子信息系列精品教材

数字电子技术

主　编　马明涛　张　杰
副主编　李广羽　王丽芬　王冬雪

中国矿业大学出版社
·徐　州·

图书在版编目（CIP）数据

数字电子技术 / 马明涛，张杰主编. -- 徐州 : 中国矿业大学出版社，2026. 1. -- ISBN 978-7-5646-7140-2

Ⅰ. TN79

中国国家版本馆CIP数据核字第2026EP6661号

书　　名　数字电子技术

主　　编　马明涛　张　杰

责任编辑　齐　畅

责任校对　张梦瑶

出版发行　中国矿业大学出版社有限责任公司

（江苏省徐州市解放南路　邮编221008）

电　　话　（0516）83884920（总编办）　83885105（营销部）

网　　址　http: //www.cumtp.com　**E-mail**: cumtpvip@cumtp.com

印　　刷　三河市骏杰印刷有限公司

开　　本　787 mm×1092 mm　1/16　**印张**　18　**插页**　2　**字数**　408千字

版次印次　2026年1月第1版　2026年1月第1次印刷

书　　号　ISBN 978-7-5646-7140-2

定　　价　59.90元

前言 PREFACE

本书主要依据教育部高等学校电工电子基础课程教学指导分委员会制定的《高等学校电子电气基础课程教学基本要求（2024 年版）》编写。随着数字电子技术的高速发展，该领域里的新概念、新器件和新方法不断涌现，使数字电子技术基础课程的教学内容不断增加，教材篇幅过大。而随着教学改革的不断深入、培养方案的不断调整，各门课程的授课学时反而在减少，这样就加剧了数字电子技术基础课程所需学时多而实际授课学时少的矛盾，于是我们编写了这本实用性强、所用学时少的教材。

本书紧密围绕党的二十大精神编写。在内容编排上，特别注重培养学生的创新思维与实践能力。书中不仅系统阐述了数字电子技术的基础理论，还引入了大量数字集成器件的应用案例，让学生深入了解学科发展趋势，激发学生的创新热情。本书精心设计了一系列例题和课后习题，鼓励学生运用所学知识解决实际问题，力求在实践中提升创新能力与技术应用水平。

本书的编写融入了编者多年的教学实践经验。为了有效地实现课程内容整合，编者在内容的选取和衔接、课后习题的选编等方面做了深入探讨，最终形成了一定的编写原则。本书主要内容包括数字电路基础、集成逻辑门电路、组合逻辑电路、触发器、时序逻辑电路、脉冲波形发生器与整形电路、数 / 模和模 / 数转换器、半导体存储器和可编程逻辑器件、硬件描述语言等。本书的特点如下：

（1）从数字电子技术基础课程的定位出发，既满足教学内容的需要，又具有一定的实用性。

（2）内容以流行的集成电路为主，适当保留了门电路和触发器方面的知识，把分析和设计的重点从门电路和触发器转移到集成电路。

（3）减少了集成电路内部工作原理的介绍，强调通过外特性来学习集成电路。

（4）适当引入了新概念、新器件和新技术，便于学生了解电子技术的新发展并掌握分析和设计数字电路的新方法。

本书适用于高等院校电气信息类、机电类和仪器仪表类专业的学生。本书建议教学56学时，其中理论教学48学时，实验教学8学时，可根据具体教学计划调整。关于理论教学，第1章建议分配8学时，第2章建议分配2学时，第3章建议分配10学时，第4章建议分配8学时，第5章建议分配10学时，第6 ~ 9章建议分配10学时。本书编者多是授课多年的一线教师，具有丰富的教学经验。本书由吉林农业科技学院马明涛、张杰任主编，吉林农业科技学院李广羽、王丽芬和吉林工程职业学院王冬雪任副主编。具体编写分工如下：王丽芬编写第1、9章；李广羽编写第2章；张杰编写第3、7、8章；马明涛和王冬雪编写第4、5、6章，并负责制定编写提纲和全书的统稿工作。另外，参加本书编写的还有吉林农业科技学院左震、李鹏、咸悦、刘志东、王美涵，珠海世纪鼎利科技股份有限公司黄伟，吉林省经济管理干部学院陈芸婧，吉林化工大学研究生叶琪、赵鑫。在此向所有支持本书编写的同人一并表示感谢。

由于编者水平有限，书中疏漏和不足之处在所难免，欢迎广大读者提出宝贵意见和建议，以便再版时修订完善。

编　者

2025年4月

目 录 CONTENTS

第 3 章　组合逻辑电路

第 4 章　触发器

第 5 章　时序逻辑电路

第 6 章　脉冲波形发生器与整形电路

第 7 章　数 / 模和模 / 数转换器

第 1 章

数字电路基础

本章导读

常用数制、码制及数制之间的转换。

常用的基本逻辑运算及其表示方法。

逻辑函数表示方法及其转换。

逻辑代数的基本公式、定律和常用规则。

逻辑函数的代数化简法和卡诺图化简法。

正、负逻辑的概念及其转换。

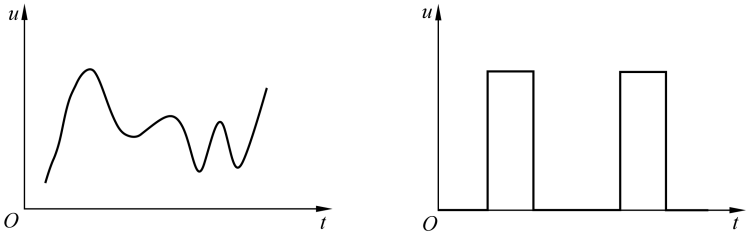
本章首先介绍数字电路的分类和特点，并从十进制数的运算规则引入二进制、八进制、十六进制数的运算规则及它们之间的相互转换；接着介绍逻辑代数中的基本逻辑运算、基本公式、基本定律和规则，这对逻辑函数的化简是十分有用的；最后介绍逻辑函数的几种常用表示方法及逻辑函数的代数化简法和卡诺图化简法。

1.1 数字电路概述

1.1.1 信号和电路的分类

电子系统中的信号可以分为两大类：模拟信号和数字信号（图 1–1）。模拟信号是时间连续、数值也连续的信号，如模拟广播电视传送的音频信号和视频信号等。数字信号是在时间上和数值上均离散的信号，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。具有对模拟信号进行放大、滤波、调制、解调、传输等的处理能力的电路称为模拟电路；能够对数字信号进行采集、存储、传输、变换、运算及显示等处理的电路

是数字电路。数字电路主要研究输出信号与输入信号之间的对应逻辑关系，其分析的主要工具是逻辑代数，因此数字电路又被称为“逻辑电路”。



(a) 模拟信号　　(b) 数字信号

图 1–1　模拟信号和数字信号示意图

1.1.2　数字电路的特点

数字电路具有以下一些特点。

（1）便于高度集成化。数字电路采用二进制数，仅有 0 和 1 两种状态，对电路元器件的参数和精度要求不高，基本单元电路的结构简单，便于电路的集成化。

（2）工作可靠性高，抗干扰能力强。数字电路用 1 和 0 来表示信号的有和无，容易辨别，从而大大提高了电路的工作可靠性。同时，只要外界干扰在电路的噪声容限范围内，电路都能正常工作，因此抗干扰能力强。

（3）便于长期保存。比如可将数字信息存入磁盘、光盘等长期保存。

（4）产品系列多，通用性强，成本低。可采用标准的逻辑部件和可编程逻辑器件来实现各种各样的电路和系统，使用灵活。

（5）保密性好。可以采用多种编码技术加密数字信息，使其不易被窃取。

（6）具有“逻辑思维”能力。数字电路不仅具有算术运算能力，而且能按人们设计的规则进行逻辑推理和逻辑判断。

数字电路由于具有上述特点，因而发展十分迅速，在电子计算机、数控技术、通信技术、数字仪表等领域都得到了越来越广泛的应用。

1.1.3　数字电路的分类

数字电路具有以下一些分类方法。

（1）按结构不同，可把数字电路分为分立元件电路和集成电路。分立元件电路是将晶体管、电阻和电容等元器件用导线在线路板上连接而成的电路。集成电路（图 1–2）则是将元器件和导线通过半导体制造工艺做在一块硅片上，从而成为一个不可分割的整体电路。

图 1–2　集成电路

（2）根据集成度不同，可把集成电路分为 4 类，见表 1–1。这里的集成度是指组成集成电路的逻辑门或元件个数。

表 1-1　集成电路的分类

类　　型	集成度	电路规模与范围
小规模集成电路（small scale integrated circuit，SSI）	1 ~ 10 个逻辑门 / 片或 10 ~ 100 个元件 / 片	逻辑单元电路、逻辑门电路及集成触发器等
中规模集成电路（medium scale integrated circuit，MSI）	10 ~ 100 个逻辑门 / 片或 100 ~ 1 000 个元件 / 片	逻辑部件、译码器、计数器及比较器等
大规模集成电路（large scale integrated circuit，LSI）	100 ~ 1 000 个逻辑门 / 片或 1 000 ~ 10 000 个元件 / 片	数字逻辑系统、控制器、存储器及接口电路等
超大规模集成电路（very large scale integrated circuit，VLSI）	大于 1 000 个逻辑门 / 片或大于 10 000 个元件 / 片	高集成度数字逻辑系统及单片机等

（3）按所集成的元件不同，可把数字电路分为双极型电路［典型代表是晶体管 – 晶体管逻辑（transistor–transistor logic，TTL）电路］和单极型电路［典型代表是互补金属氧化物半导体（complementary metal–oxide–semiconductor，CMOS）电路］两种。

（4）按电路工作原理不同，可把数字电路分为组合逻辑电路和时序逻辑电路两种。关于这两种电路的特点和具体电路将在后面的章节详细介绍。

1.1.4　数字电路的应用

如今，数字电路已被广泛应用于计算机、自动化装置、医疗仪器与设备、交通（如交通灯等）、电信（如卫星通信等）、文娱活动等几乎所有的生产生活领域中，可以毫不夸张地说，几乎所有人每天都在与数字电路打交道。

然而，数字电路的应用也具有其局限性。因为被控制和被测量的对象往往是一些模拟信号，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，必须通过专门的电路将它们转换为数字信号（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（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，而且使信号的精度受到影响。因此，在使用数字电路时，应具体情况具体分析，以达到便于操作、提高生产效率的目的。

1.2　数制和码制

1.2.1　数制

为了描述数的大小或多少，人们采用进位计数的方法，称为进位计数制，简称数制。组成数制的两个基本要素是进位基数与数位权值，简称基数与位权。

基数：一个数位上可能出现的基本数码的个数，记为 R。

位权：基数的幂，记为 R^i，它与数码在数中的位置有关。

以日常生活中最常见的十进制数为例。十进制有 10 个数码（0、1、2、3、4、5、6、7、8、9），则基数 $R=10$，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关系是“逢十进一”。十进制数 $137=1\times10^2+3\times10^1+7\times10^0$，$10^2$、$10^1$、$10^0$ 分别为最高位、中间位和最低位的位权。除了最常见的十进制，还有其他数制，如二进制、八进制、十六进制等，同一串数字，数制不同，代表的数值大小也不同。

1. 常用数制及其表示方法

1）十进制

十进制基数 $R=10$，有 0 ~ 9 十个数码，进位规则是逢 10 进 1，各位的权值为 10^i。

任意一个十进制数 $(D)_{10}$，都可表示为

$$(D)_{10}=k_n10^{n-1}+k_{n-1}10^{n-2}+\cdots+k_110^0+k_010^{-1}+\cdots+k_{-m}10^{-m-1} \qquad (1\text{–}1)$$

式中，k_i 表示 0 ~ 9 十个数码中的任意一个；m、n 表示正整数；10 表示十进制的基数。

例如，$(2001.9)_{10}=2\times10^3+0\times10^2+0\times10^1+1\times10^0+9\times10^{-1}$。

十进制是人们最熟悉的数制，但不适合在数字电路系统中应用。

2）二进制

二进制基数 $R=2$，有 0、1 两个数码，进位规则是逢 2 进 1，各位的权值是 2^i。

任意一个二进制数 $(D)_2$，都可表示为

$$(D)_2=k_n2^{n-1}+k_{n-1}2^{n-2}+\cdots+k_12^0+k_02^{-1}+\cdots+k_{-m}2^{-m-1} \qquad (1\text{–}2)$$

式中，k_i 表示 0 或 1；m、n 表示正整数；2 表示二进制的基数。

例如，$(1101.101)_2=1\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}$。

二进制计数规则简单，存储、传递方便，广泛应用于数字系统，但对于较大的数值，需要较多位表示，书写太长，不够方便。

3）八进制

八进制基数 $R=8$，有 0 ~ 7 八个数码，进位规则是逢 8 进 1，各位的权值是 8^i。

任意一个八进制数 $(D)_8$，都可表示为

$$(D)_8=k_n8^{n-1}+k_{n-1}8^{n-2}+\cdots+k_18^0+k_08^{-1}+\cdots+k_{-m}8^{-m-1} \qquad (1\text{–}3)$$

式中，k_i 表示 0 ~ 7 八个数码中的任意一个；m、n 表示正整数；8 表示八进制的基数。

例如，$(67.731)_8=6\times8^1+7\times8^0+7\times8^{-1}+3\times8^{-2}+1\times8^{-3}$。

4）十六进制

十六进制基数 $R=16$，有 0 ~ 9、A ~ F 十六个数码，进位规则是逢 16 进 1，各位的权值是 16^i。

任意一个十六进制数 $(D)_{16}$，都可表示为

$$(D)_{16}=k_n16^{n-1}+k_{n-1}16^{n-2}+\cdots+k_116^0+k_016^{-1}+\cdots+k_{-m}16^{-m-1} \quad (1\text{–}4)$$

式中，k_i 表示 0 ~ 9、A ~ F 十六个数码中的任意一个；m、n 表示正整数；16 表示十六进制的基数。

例如，$(8AE6)_{16}=8\times16^3+A\times16^2+E\times16^1+6\times16^0=8\times16^3+10\times16^2+14\times16^1+6\times16^0$。

5）任意进制（R 进制）

R 进制的基数为 r，有 0 ~（$r-1$）个数码，一般表示为

$$(D)_r=k_nr^{n-1}+k_{n-1}r^{n-2}+\cdots+k_1r^0+k_0r^{-1}+\cdots+k_{-m}r^{-m-1} \quad (1\text{–}5)$$

式中，k_i 表示 0 ~（$r-1$）r 个数码中的任意一个；m、n 表示正整数；r 表示 R 进制的基数。

在计算机系统中，二进制主要用于机器内部的数据处理，八进制和十六进制主要用于书写程序，十进制主要用于运算最终结果的输出。

为了便于对照，将常用的几种数制之间的关系列于表 1–2 中。

表 1–2　几种常用数制及其对应关系

类　别	十进制	二进制	八进制	十六进制
表示数码	0，1，…，9	0，1	0，1，…，7	0，1，…，9；A ~ F
进位规则	逢 10 进 1	逢 2 进 1	逢 8 进 1	逢 16 进 1
第 i 位权值	10^i	2^i	8^i	16^i
对应关系	0	0	0	0
	1	1	1	1
	2	10	2	2
	3	11	3	3
	4	100	4	4
	5	101	5	5
	6	110	6	6
	7	111	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
	16	10000	20	10

2. 数制间的转换

数字系统常用的数制为十进制和二进制。十进制是人们最熟悉的数制，但硬件实现起来困难。二进制是机器唯一认识的数制，但二进制书写太长。因此，引入八进制和十六进制等数制，各种数制都有自己的应用场合，各种数制间需要转换。

1）R 进制数转换为十进制数

如果将 R 进制数转换为等值的十进制数，只需要将 R 进制数按位权展开，再按十进制运算规则运算即可。

【例 1–1】将二进制数 $(1101.11)_2$ 转换成十进制数。

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

【例 1–2】将八进制数 $(73.51)_8$ 转换成十进制数。

解：$(73.51)_8 = 7\times8^1 + 3\times8^0 + 5\times8^{-1} + 1\times8^{-2} = (59.640\,625)_{10}$

【例 1–3】将十六进制数 $(F3D.54)_{16}$ 转换成十进制数。

解：$(F3D.54)_{16} = 15\times16^2 + 3\times16^1 + 13\times16^0 + 5\times16^{-1} + 4\times16^{-2} = (3\,901.328\,125)_{10}$

2）十进制数转换为 R 进制数

将十进制数转换为 R 进制数时，需先将十进制数的整数部分和小数部分分别转换，然后将转换结果合并起来。

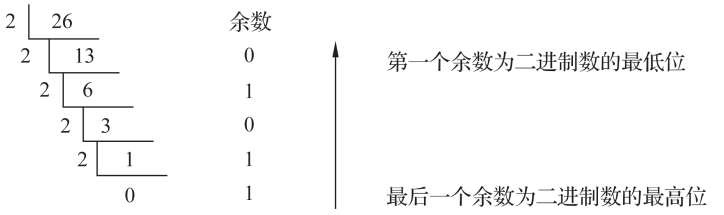
（1）十进制数转换成二进制数。

整数部分：用除以 2 取余的方法进行转换，先余为低，后余为高。

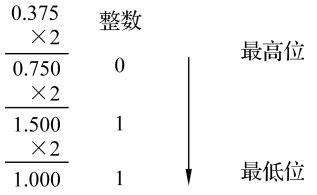
小数部分：用乘以 2 取整的方法进行转换，先整为高，后整为低。

【例 1–4】将 $(26.375)_{10}$ 转换成二进制数。

解：整数部分，



小数部分，



所以，有 $(26.375)_{10} = (11010.011)_2$。

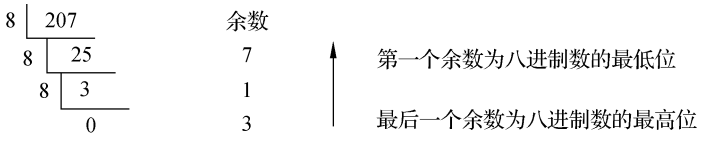
（2）十进制数转换成八进制数。

整数部分：用除以 8 取余的方法进行转换，先余为低，后余为高。

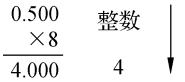
小数部分：用乘以 8 取整的方法进行转换，先整为高，后整为低。

【例 1–5】将 $(207.5)_{10}$ 转换成八进制数。

解：整数部分，



小数部分，



取出整数 4，余数为 0，转换结束。

综上可得 $(207.5)_{10}=(317.4)_8$。

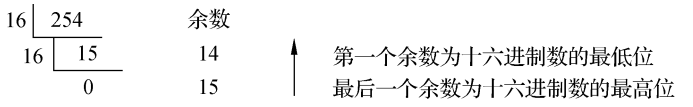
（3）十进制数转换成十六进制数。

整数部分：用除以 16 取余的方法进行转换，先余为低，后余为高。

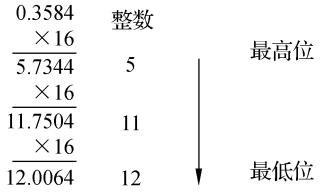
小数部分：用乘以 16 取整的方法进行转换，先整为高，后整为低。

【例 1–6】将 $(254.3584)_{10}$ 转换为十六进制数。

解：整数部分，



小数部分，



最终转换结果为：$(254.3584)_{10}=(FE.5BC)_{16}$。

3）二进制数与八进制数、十六进制数相互转换

二进制数转换成八进制数（或十六进制数）的规则如下。

从小数点算起，向左或向右每 3（或 4）位分成一组，最后不足 3（或 4）位用 0 补齐，每组用 1 位等值的八进制数（或十六进制数）表示，即得到要转换的八进制数（或十六进制数）。

【例 1–7】将 $(10111011.01111)_2$ 转换成八进制数和十六进制数。

解：二进制　010 111 011 . 011 110

八进制　2　7　3　.　3　6

所以 $(10111011.01111)_2=(273.36)_8$。

二进制 1011 1011 . 0111 1000

十六进制 B B . 7 8

所以 $(10111011.01111)_2=(BB.78)_{16}$。

反之，八进制数（或十六进制数）转换成二进制数时，只要将每位八进制数（或十六进制数）分别写成相应的 3（或 4）位二进制数，按原来的顺序排列起来即可。

利用八进制数和十六进制数与二进制数之间的这种关系，可以进行八进制数与十六进制数之间的相互转换。

1.2.2 码制

用按一定规律排列的多位二进制数码表示某种信息，称为编码。形成代码的规律法则，称为码制。

在数字系统中，二进制代码是由 0、1 构成的不同的组合，这里的“二进制”并无“进位”的含义，只是强调采用的是二进制数的数码符号而已。n 位二进制数可有 2^n 种不同的组合，即可代表 2^n 种不同的信息。

1. 二－十进制码

用 4 位二进制数码表示 1 位十进制数的代码，称为二－十进制码，简称 BCD 码（binary coded decimal）。4 位二进制数有 16 种组合，而 1 位十进制数只需要 10 种组合，因此，用 4 位二进制码表示 1 位十进制数的组合方案有许多种，无论按哪种组合方案，均有 6 种组合的冗余。表 1–3 列出了几种常用的 BCD 码。

表 1–3 几种常用的 BCD 码

十进制数	编码			
	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

在表 1–3 中，8421 码、2421 码、5421 码都属于有权码，而余 3 码属于无权码。

1）8421 码

8421 码是最常用的一种 BCD 码，它与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 十个数码，这 10 个组合称为有效码；余下的 6 个组合 1010 ~ 1111 没有被采用，称为无效码。

8421 码是一种有权码，因而根据代码的组成便可知道它所代表的值。设 8421 码的各位为 a_3、a_2、a_1、a_0，则它所代表的值为

$$N = 8a_3 + 4a_2 + 2a_1 + 1a_0 \qquad (1\text{–}6)$$

8421 码编码简单直观，能很容易地实现 8421 码到十进制数的转换。8421 码与十进制数之间的转换只要直接按位转换即可，例如，

$$(509.37)_{10} = (0101\ 0000\ 1001.\ 0011\ 0111)_{8421}$$

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，舍弃头、尾 3 个组合而形成。因此，余 3 码所代表的十进制数可由下式算得：

$$N = 8a_3 + 4a_2 + 2a_1 + 1a_0 - 3 \qquad (1\text{–}7)$$

式中，a_3、a_2、a_1、a_0 为余 3 码的各位数（0 或 1）。

余 3 码是一种无权代码，该代码中的各位“1”不表示一个固定值，因而不直观，且容易搞错。余 3 码也是一种自反代码。例如，4 的余 3 码为 0111，将它的各位取反得 1000，即 5 的余 3 码，而 4 与 5 对 9 互反。

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，

$$(85.93)_{10} = (1011\ 1000.\ 1100\ 0110)_{余3}$$

不同的编码各有优缺点，在数字电路设计中，我们要根据设计需要采用不同的码制。

2. 可靠性编码

代码在产生和传输过程中，难免发生错误。为减少错误发生，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。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 1–4 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 1–4　典型的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
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

在时序电路中采用格雷码编码，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换。例如，8421 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（counter）］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，且其他位未变为 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错误。

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。奇偶校验码的实现方法是在每个被传送码的左边或右边加上 1 位奇偶校验位“0”或“1”，使得整个码组中 1 的数目为奇数或者为偶数。若为奇数，称为奇校验码；若为偶数，称为偶校验码。表 1–5 是 8421 码加了奇校验位和偶校验位之后的编码，表中的校验码加在了信息码的右边。

表 1-5　8421 码的奇偶校验码

十进制数	信息码	带奇校验位的 8421 码	带偶校验位的 8421 码
0	0000	00001	00000
1	0001	00010	00011
2	0010	00100	00101
3	0011	00111	00110
4	0100	01000	01001
5	0101	01011	01010
6	0110	01101	01100
7	0111	01110	01111
8	1000	10000	10001
9	1001	10011	10010

1.3 基本逻辑运算

1.3.1 逻辑函数和逻辑变量

1. 逻辑函数

在研究事件的因果关系时，决定事件变化的因素称为逻辑自变量，对应事件的结果称为逻辑因变量，也叫逻辑结果。以某种形式表示逻辑自变量与逻辑结果之间的函数关系称为逻辑函数。例如，当逻辑自变量 A，B，C，D，…的取值确定后，逻辑因变量 Y 的取值也就唯一确定了，则称 Y 是 A，B，C，D，…的逻辑函数，记作 $Y = f(A, B, C, D, \cdots)$ 。

2. 逻辑变量

逻辑代数中的变量称为逻辑变量，逻辑变量分为两类，即输入逻辑变量和输出逻辑变量。无论是输入逻辑变量还是输出逻辑变量，它们的取值都只有两种，即 0 和 1。区别于数字变量，这里的 0 和 1 并没有数的含义，它们表示两种完全对立的逻辑状态。例如，若用 1 表示开关闭合，则 0 表示开关断开；若用 1 表示电灯亮，则 0 表示电灯灭；若用 1 表示高电平，则 0 表示低电平；等等。

1.3.2 三种基本逻辑关系及其表示方法

基本的逻辑关系有与逻辑、或逻辑和非逻辑三种，与它们对应的逻辑运算分别为与运算（逻辑乘）、或运算（逻辑加）和非运算（逻辑非或逻辑反）。

1. 与逻辑

当决定某个事件的全部条件都具备时，才发生该事件，这种因果关系称为“与逻辑”。

例如，串联开关控制的照明电路即为与逻辑（图 1–3）。

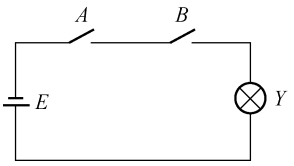


图 1–3　串联开关控制的照明电路

在图 1–3 中，开关 A 和 B 的状态（闭合或断开）与电灯 Y 的状态（亮和灭）之间存在确定的因果关系。显然只有当串联的两个开关都闭合时，灯才能亮。如果规定开关闭合为逻辑 1 态，开关断开为逻辑 0 态，灯亮为逻辑 1 态，灯灭为逻辑 0 态，则开关 A 和 B 的全部状态组合与灯 Y 状态之间的关系见表 1–6。

表 1–6　与逻辑的真值表

输　入		输　出	输出特点
A	B	Y	
0	0	0	有 0 出 0
0	1	0	
1	0	0	
1	1	1	全 1 出 1

上述逻辑关系可用式（1–8）表示：

$$Y = A \cdot B \tag{1–8}$$

多变量的与逻辑关系可用式（1–9）表示：

$$Y = A \cdot B \cdot C \cdot \cdots \tag{1–9}$$

式中，“·”表示逻辑乘，又称为“与逻辑运算”，实现与逻辑运算的电路称为“与门”。在不需要强调的地方，“·”可省略。

国际标准使用的与门逻辑符号如图 1–4 所示。

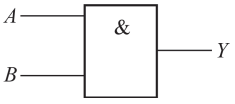


图 1–4　与门逻辑符号

2. 或逻辑

当决定某个事件的全部条件中有一个或一个以上条件具备时，就发生该事件，这种因果关系称为“或逻辑”。例如，图 1–5 所示的并联开关控制的照明电路即为一个简单的或逻辑。

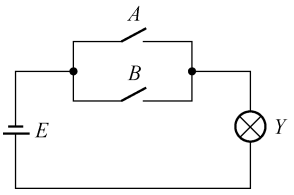


图 1-5　并联开关控制的照明电路

或逻辑的真值表如表 1-7 所示。

表 1-7　或逻辑的真值表

输　入		输　出	输出特点
A	B	Y	
0	0	0	全 0 出 0
0	1	1	有 1 出 1
1	0	1	
1	1	1	

上述逻辑关系可用式（1-10）表示：

$$Y = A + B \tag{1-10}$$

多变量的或逻辑关系可用式（1-11）表示：

$$Y = A + B + C + \cdots \tag{1-11}$$

式中，“+”表示逻辑加，又称为“或逻辑运算”，实现或逻辑运算的电路称为“或门”。

国际标准使用的或门逻辑符号如图 1-6 所示。

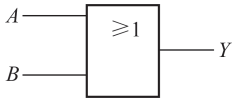


图 1-6　或门逻辑符号

3. 非逻辑

非逻辑也称为“逻辑反”，数字电路中的反相器就是实现非逻辑的电子元件，在现实中经常使用。

决定某一事件的条件满足时，事件不发生；反之，事件发生。

例如，图 1-7 所示的开关与负载并联的控制电路即为非逻辑的一个实际应用。

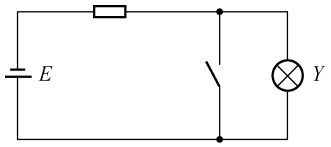


图 1-7　开关与负载并联的控制电路

非逻辑的真值表如表 1–8 所示。

表 1–8　非逻辑的真值表

输　入	输　出	输出特点
A	Y	
0	1	有 0 出 1
1	0	有 1 出 0

非逻辑表达式为

$$Y = \overline{A} \tag{1–12}$$

在变量上方的“¯”号表示非，$\overline{A}$ 读作“A 非”。显然，A 和 $\overline{A}$ 互为反变量。实现非运算的电路称为“非门”，由于非门的输出信号与输入信号反相，故“非门”又被称为“反相器”。

国际标准使用的非门逻辑符号如图 1–8 所示。

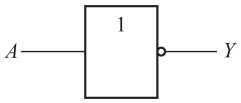


图 1–8　非门逻辑符号

1.4　复合逻辑运算

与、或、非是三种基本逻辑运算，实际的逻辑问题往往比与、或、非复杂得多，不过这些复杂的逻辑运算都可以通过三种基本的逻辑运算组合而成。最常见的复合逻辑运算有与非运算、或非运算、异或运算、同或运算以及与或非运算。它们所对应的逻辑门分别是与非门、或非门、异或门、同或门以及与或非门。其逻辑表达式、逻辑符号、功能特征及真值表如表 1–9、表 1–10 所示。

表 1–9　几种常见的复合逻辑运算

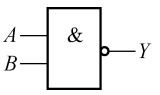
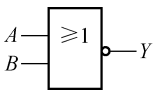
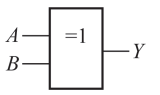
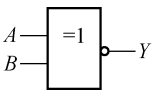
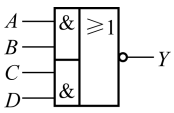
逻辑关系	与　非	或　非	异　或	同　或	与或非
逻辑表达式	$Y = \overline{A \cdot B}$	$Y = \overline{A + B}$	$Y = \overline{A}B + A\overline{B}$ $= A \oplus B$	$Y = \overline{A}\overline{B} + AB$ $= A \odot B$	$Y = \overline{AB + CD}$
逻辑符号					
功能特征	所有输入均为 1 时，输出为 0	所有输入均为 0 时，输出为 1	两个输入相异时，输出为 1	两个输入相同时，输出为 1	所有与项为 0 时，输出为 1

表 1-9（续）

逻辑关系			与　非	或　非	异　或	同　或	与或非
真值表	A	B	Y	Y	Y	Y	见表 1-10
	0	0	1	1	0	1	
	0	1	1	0	1	0	
	1	0	1	0	1	0	
	1	1	0	0	0	1	

表 1-10　与或非运算真值表

输　入				输　出	输　入				输　出
A	B	C	D	Y	A	B	C	D	Y
0	0	0	0	1	1	0	0	0	1
0	0	0	1	1	1	0	0	1	1
0	0	1	0	1	1	0	1	0	1
0	0	1	1	0	1	0	1	1	0
0	1	0	0	1	1	1	0	0	0
0	1	0	1	1	1	1	0	1	0
0	1	1	0	1	1	1	1	0	0
0	1	1	1	0	1	1	1	1	0

1.5　逻辑函数的表示方法及其相互转换

常用的逻辑函数表示方法有逻辑真值表（简称真值表）、逻辑表达式（也称逻辑式或逻辑函数式）、逻辑图、波形图、卡诺图等。

将输入变量的所有取值及对应的输出值，以表格的形式一一列举出来，即可得到真值表。每一个输入变量有 0、1 两个取值，对于一个逻辑电路，若有 n 个输入变量，则 n 个变量各种可能取值的组合有 2^n 种，其对应逻辑函数输出值就有 2^n 个，如前面各种逻辑运算的真值表。真值表可直观地反映出输出与输入的因果关系。

所谓逻辑表达式，是指用与、或、非等逻辑运算的组合形式所表示的输入输出逻辑变量之间关系的逻辑代数式。逻辑表达式表示方法简捷，也便于利用代数法对其进行化简。如逻辑函数 $Y = AB + \overline{B}CD$。

逻辑图是指用基本和常用的逻辑符号表示函数表达式中各个变量之间的运算关系的图形。逻辑图和逻辑表达式之间有着严格的一一对应关系，它们之间的互相转换比较方便。但是逻辑图和真值表一样，也不能直接运用公式和定理进行运算和变换。逻辑图是电路设计结果的表现形式。

波形图是指输入变量和对应的输出变量随时间变化的图形。波形图能够形象直观地表示变量取值与函数值在时间上的对应关系，但难以用公式和定理进行运算与变换，当变量

个数增多时，画图较麻烦。

卡诺图是真值表的一种方块图表达形式，要求变量取值必须按照循环码的顺序排列，便于求出逻辑函数的最简与或表达式。卡诺图只适于表示和化简变量个数比较少的逻辑函数，也不便于进行运算和变换。

同一个逻辑关系可以用不同的表示方法来表示，各种表示方法之间均可以进行直接或间接的变换。

1.5.1 已知真值表求逻辑表达式和逻辑图

由真值表求逻辑表达式的一般方法如下。

（1）找出使逻辑函数 $Y=1$ 的行，每一行用一个乘积项表示，其中变量取值为“1”时用原变量表示，变量取值为“0”时用反变量表示。

（2）将所有的乘积项进行或运算，即可以得到 Y 的逻辑表达式。

由真值表求逻辑图的一般方法如下。

（1）根据真值表写出逻辑函数的逻辑表达式。

（2）对逻辑表达式进行化简（化简的方法将在 1.6 节中详细介绍）。

（3）把逻辑表达式中各个变量之间的逻辑运算用相应的逻辑符号表示出来，就得到了对应的逻辑图。

【例 1–8】已知一个函数的真值表（表 1–11），试写出它的逻辑表达式并画出逻辑图。

表 1–11　例 1–8 的真值表

输　入			输　出
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

解：在表中查到，使函数 Y 为 1 的变量取值组合为

A=0，B=1，C=1；

A=1，B=0，C=1；

A=1，B=1，C=0；

A=1，B=1，C=1。

得到乘积项为 $\overline{A}BC$、$A\overline{B}C$、$AB\overline{C}$ 和 ABC，将这四个乘积项相加，得到的逻辑表达式为

$$Y=\overline{A}BC+A\overline{B}C+AB\overline{C}+ABC=AB+AC+BC。$$

有了逻辑表达式，按照先后顺序，用逻辑符号表示并正确连接起来就可以画出如图 1–9 所示的逻辑图。

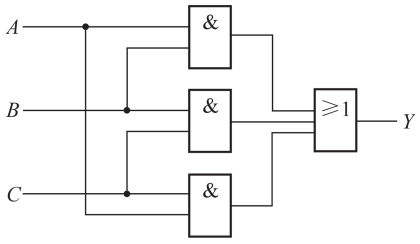


图 1–9　例 1–8 的逻辑图

1.5.2　已知逻辑表达式求真值表和逻辑图

如果有了逻辑表达式，则只要先把输入变量取值的所有组合状态逐一代入函数中算出逻辑函数值，然后将输入变量取值与逻辑函数值对应地列成表，就得到逻辑函数的真值表。

【例 1–9】已知逻辑表达式 $Y=\overline{A}B+\overline{A}\overline{B}C$，求与它对应的真值表和逻辑图。

解：观察逻辑表达式中有 A、B、C 三个输入变量，因此它们的各种可能取值有 $2^3=8$ 组，将每组取值一一代入逻辑表达式，求出对应 Y 值，列成表格，即得其真值表，如表 1–12 所示。

表 1–12　例 1–9 的真值表

输　入			输　出
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

根据逻辑表达式 $Y=\overline{A}B+\overline{A}\overline{B}C$，得到的逻辑图如图 1–10 所示。

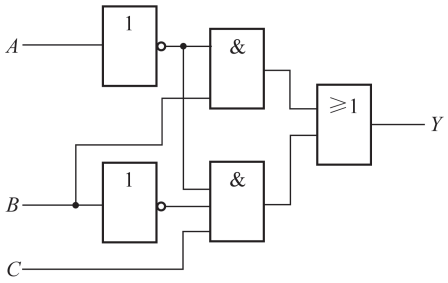


图 1–10　例 1–9 的逻辑图

1.5.3 已知逻辑图求逻辑表达式、真值表和波形图

如果只给出逻辑图，也能得到对应的逻辑表达式和真值表。逻辑表达式的转换步骤为：从输入端到输出端逐级写出输出端的逻辑表达式。有了逻辑表达式，列真值表就不难了。

【例 1–10】试写出图 1–11（a）所示逻辑图的逻辑表达式并列出其真值表。若输入端 A、B 和 C 的波形如图 1–11（b）所示，绘出输出端 Y 的波形。

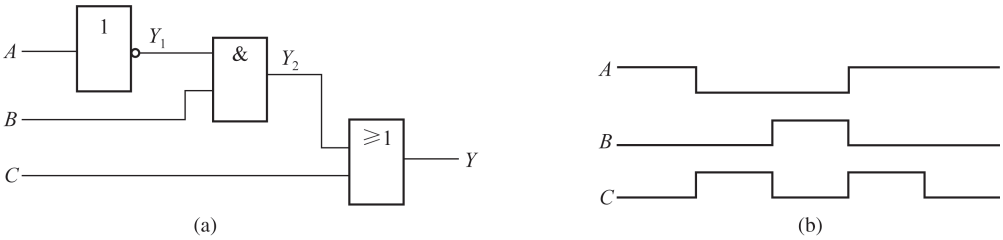


图 1–11　例 1–10 的逻辑图及波形图

解：由图 1–11（a）知 $Y_1 = \overline{A}$，$Y_2 = \overline{A}B$。

故 $Y = C + Y_2 = C + \overline{A}B$。其真值表如表 1–13 所示。

表 1–13　例 1–10 的真值表

输　入			输　出
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

已知输入端波形，可以根据逻辑表达式计算输出值，绘出输出端波形，或者根据真值表查得输出值，绘出输出端波形。例 1–10 输出端波形如图 1–12 所示。

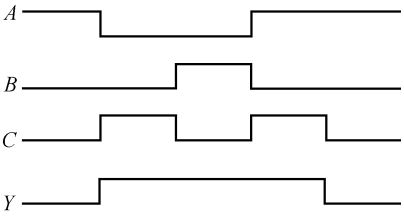


图 1–12　例 1–10 的波形图

1.6　逻辑代数

逻辑代数是一种用于描述客观事物逻辑关系的数学方法，由英国数学家乔治·布尔（George Boole）于 1847 年提出，因而又被称为布尔代数。逻辑代数和普通代数一样，有一套完整的运算规则，包括公理、定理和定律。它被广泛地应用于开关电路和数字逻辑电路的变换、分析、化简和设计上，因此也被称为开关代数。逻辑代数是研究逻辑电路的数学工具，它为分析和设计逻辑电路提供了理论基础。根据三种基本逻辑运算，可推导出一些基本公式和定律，形成一些基本运算规则。熟悉、掌握并且会运用这些规则，对于掌握数字电子技术十分重要。

1.6.1　基本公式、定律和常用规则

1. 逻辑代数的基本公式、定律

逻辑代数的基本公式和定律如表 1–14 所示。

表 1–14　逻辑代数的基本公式和定律

定　律	公　式	
0-1 律	$\overline{0}=1$ $0\cdot A=0$ $1\cdot A=A$	$\overline{1}=0$ $1+A=1$ $0+A=A$
交换律	$A\cdot B=B\cdot A$	$A+B=B+A$
结合律	$A\cdot(B\cdot C)=(A\cdot B)\cdot C$	$A+(B+C)=(A+B)+C$
分配律	$A\cdot(B+C)=A\cdot B+A\cdot C$	$A+B\cdot C=(A+B)\cdot(A+C)$
吸收律	$A\cdot(A+B)=A$	$A+A\cdot B=A$
重复律	$A\cdot A=A$	$A+A=A$
互补律	$A\cdot\overline{A}=0$	$A+\overline{A}=1$
还原律	$\overline{\overline{A}}=A$	
反演律[①]	$\overline{A\cdot B}=\overline{A}+\overline{B}$	$\overline{A+B}=\overline{A}\cdot\overline{B}$

注：反演律又叫德·摩根（De Morgan）定律，在逻辑函数的化简及变换中经常用到。

2. 逻辑代数的常用公式

利用表 1–14 中的运算定律可以得到更多的公式。

公式 1　　$A+\overline{A}B=A+B$　　（1–13）

证明：$A+\overline{A}B=(A+\overline{A})(A+B)=A+B$。

公式 1 的含义：两个乘积项相加时，如果一项取反后是另一项的因子，则此因子是多余的，可以消去。

公式 2　　$AB+A\overline{B}=A$　　（1–14）

证明：$AB+A\overline{B}=A(B+\overline{B})=A$。

公式 2 的含义：在与或表达式中，若两个项中分别包含了一个变量的原变量和反变量，而其余因子又相同，则可合并成一项，保留其相同的因子。

公式 3　　$AB+\overline{A}C+BC=AB+\overline{A}C$，$AB+\overline{A}C+BCD=AB+\overline{A}C$　　（1–15）

证明：$AB+\overline{A}C+BC=AB+\overline{A}C+BC(A+\overline{A})=AB+\overline{A}C+ABC+\overline{A}CB=AB+\overline{A}C$，

$AB+\overline{A}C+BCD=AB+\overline{A}C+BC+BCD=AB+\overline{A}C+BC=AB+\overline{A}C$。

公式 3 的含义：在一个与或表达式中，一个与项包含了一个变量的原变量，而另一个与项包含了这个变量的反变量，则这两个与项中其余因子的乘积构成的第三项是多余的，可以消去。因此，这两个公式也叫冗余律。

公式 4　　$\overline{\overline{A}B+A\overline{B}}=\overline{A}\overline{B}+AB$　　（1–16）

证明：由反演律得

$\overline{\overline{A}B+A\overline{B}}=(\overline{\overline{A}B})\cdot(\overline{A\overline{B}})=(A+\overline{B})(\overline{A}+B)=A\overline{A}+AB+\overline{A}\overline{B}+B\overline{B}=\overline{A}\overline{B}+AB$。

由于 $A\oplus B=\overline{A}B+A\overline{B}$，$A\odot B=\overline{A}\overline{B}+AB$，

所以式（1–16）可写为

$\overline{A\oplus B}=A\odot B$。　　（1–17）

3. 逻辑代数的常用规则

1）代入规则

在任一个逻辑等式中，若将等式两边出现的某变量 A 都用同一个逻辑函数替代，且替代后等式仍然成立，这个规则称为“代入规则”。

代入规则的正确性是由逻辑变量和逻辑函数值的二值性保证的，因为逻辑变量只有 0 和 1 两种取值，无论将 $A=0$ 或 $A=1$ 代入逻辑等式，等式一定成立；而逻辑函数值也只有 0 和 1 两种取值，所以用它替代逻辑等式中的变量 A 后，等式仍成立。

代入规则在推导公式中有很大用途，因为将已知等式中的某一变量用任一函数代替后都可以得到一个新的等式，所以扩大了等式的应用范围。

【例 1–11】已知 $\overline{A\cdot B}=\overline{A}+\overline{B}$，试证明 $\overline{A\cdot B\cdot C}=\overline{A}+\overline{B}+\overline{C}$。

证明：将 $\overline{A\cdot B}=\overline{A}+\overline{B}$ 中两边的变量 B 都用同一个函数 $f(B,C)=B\cdot C$ 替代得：

$\overline{A\cdot f(B,C)}=\overline{A}+\overline{f(B,C)}$，

$\overline{A\cdot f(B,C)}=\overline{A\cdot(B\cdot C)}=\overline{A}+\overline{B\cdot C}$，

$\overline{A\cdot B\cdot C}=\overline{A}+\overline{B\cdot C}=\overline{A}+\overline{B}+\overline{C}$。

这个例子证明了德·摩根定律的一个推广等式，另一个等式可用类似的方法证明。

2）反演规则

对任何一个逻辑表达式 Y，如果将式中所有的“·”换成“+”，“+”换成“·”，“0”换成“1”，“1”换成“0”，原变量换成反变量，反变量换成原变量，则可得到原来逻辑表达式 Y 的反函数 $\overline{Y}$，这种变换规则称为“反演规则”。

在应用反演规则变换时必须注意下面的问题。

（1）不能改变原来的运算顺序，变换后的运算顺序要保持变换前的运算优先顺序，必要时可加括号表明运算的顺序。

（2）反变量换成原变量只对单个变量有效，而与非及或非等运算的长非号则保持不变。

【例 1–12】已知逻辑函数 $Y = A \cdot \overline{B + C} + CD$，试用反演规则求反函数 $\overline{Y}$。

解：根据反演规则，可写出

$$\begin{aligned}\overline{Y} &= \overline{A \cdot \overline{B + C} + CD} \\ &= (\overline{A} + \overline{\overline{B} \cdot \overline{C}})(\overline{C} + \overline{D}) \\ &= (\overline{A} + B + C)(\overline{C} + \overline{D}) \\ &= \overline{A}\overline{C} + \overline{A}\overline{D} + B\overline{C} + B\overline{D} + C\overline{D}\end{aligned}$$

反演规则的意义在于利用它可以比较容易地求出一个逻辑函数的反函数。

利用德·摩根定律也可求一个逻辑函数的反函数，它只是反演规则的一个特例。我们只需要先对原逻辑函数等式两边同时求反，然后用德·摩根定律变换即可。

3）对偶规则

对任何一个逻辑表达式 Y，如果将式中所有的“·”换成“+”，“+”换成“·”，“0”换成“1”，“1”换成“0”，这样会得到一个新的逻辑表达式 Y'。Y 和 Y' 是互为对偶式，这种变换规则称为“对偶规则”。

对偶变换要注意保持变换前运算的优先顺序不变。

【例 1–13】已知下列逻辑表达式，求其相应的对偶式。

$$Y_1 = \overline{\overline{A} + \overline{B + \overline{C}}} \qquad Y_2 = \overline{\overline{A} \cdot \overline{B \cdot \overline{C}}}$$

解：根据对偶规则，可写出：

$Y_1' = \overline{\overline{A} \cdot \overline{B \cdot \overline{C}}}$；　$Y_2' = \overline{\overline{A} + \overline{B + \overline{C}}}$。

对偶规则的意义在于若两个函数式相等，则其对偶式也一定相等。因此对偶规则也适用于逻辑等式，如将逻辑等式两边同时进行对偶变换，得到的对偶式仍然相等。

利用对偶规则，可以把基本逻辑定律和公式扩展一倍。在表 1–14 逻辑代数的基本公式和定律中，我们仅需记忆左边一列等式，因为根据对偶规则可以很容易地求出右边一列等式。

例如，在 0–1 律中有等式 $1\cdot A=A$，如果设 $Y_1=1\cdot A$，$Y_2=A$，则 $Y_1=Y_2$。Y_1 的对偶式 $Y_1'=0+A$，Y_2 的对偶式 $Y_2'=A$，根据对偶规则，$Y_1'=Y_2'$，即 $0+A=A$。这样，需要记忆的公式就少了一半。

1.6.2 逻辑函数的代数化简法

逻辑函数的代数化简法就是运用逻辑代数的公式和定理等对逻辑函数进行化简，也叫公式化简法。

1. 简化逻辑函数的意义

我们知道，同一个逻辑函数可以写成不同的表达式。用基本逻辑门电路实现某函数时表达式越简单，需用门电路的个数就越少，因而也就越经济可靠。进行逻辑设计时根据逻辑问题归纳出来的逻辑表达式往往不是最简逻辑表达式，并且可以有不同的形式，因此实现这些逻辑函数就会有不同的逻辑电路。实现逻辑函数之前，往往要先进行简化，即求出其最简逻辑表达式，然后根据最简逻辑表达式实现逻辑函数。简化和变换逻辑函数可以得到最简的逻辑表达式和所需要的形式，设计出最简捷的逻辑电路。这对于节省元器件、优化生产工艺、降低成本、提高系统的可靠性和产品在市场上的竞争力非常重要。

最简逻辑表达式有多种，最常用的有最简与或表达式（乘积项最少，每个乘积项中变量数最少）和最简或与表达式（和项最少，每个和项中变量数最少），不同类型的逻辑表达式的最简定义也不同。

2. 逻辑表达式的几种常见形式及其变换

逻辑函数的表达式不是唯一的，可以有多种形式，并且能相互变换，这种变换在逻辑分析和设计中经常用到。常见的逻辑表达式主要有与或式、与或非式、或与式、与非－与非式和或非－或非式。

例如，$Y=\overline{A}C+B\overline{C}$ 的 5 种形式分别如下。

（1）最简与或式：$Y=\overline{A}C+B\overline{C}$。

（2）对最简与或式两次求反，上面的反号不动，下面的反号用德·摩根定律，就可以得到最简与非－与非式。

$$\overline{Y}=\overline{\overline{\overline{A}C+B\overline{C}}}=\overline{\overline{\overline{A}C}\,\overline{B\overline{C}}}$$

（3）用反演规则求 $Y=\overline{A}C+B\overline{C}$ 的反函数 $\overline{Y}$ 的最简与或式，再对 $\overline{Y}$ 求反，就可以得到最简与或非式。

$$\overline{Y}=(A+\overline{C})(\overline{B}+C)=A\overline{B}+AC+\overline{C}C+\overline{C}\overline{B}=AC+\overline{B}\overline{C}$$

$$Y=\overline{\overline{Y}}=\overline{AC+\overline{B}\overline{C}}$$

（4）对最简与或非式两次用德·摩根定律，可以得到最简或与式。

$$Y=\overline{AC+\overline{B}\overline{C}}=\overline{AC}\,\overline{\overline{B}\overline{C}}=(\overline{A}+\overline{C})(B+C)$$

（5）对最简或与式两次求反，上面的反号不动，下面的反号用德·摩根定律，可以得

到最简或非－或非式。

$$Y=(\overline{A}+\overline{C})(B+C)=\overline{\overline{(\overline{A}+\overline{C})(B+C)}}=\overline{\overline{\overline{A}+\overline{C}}+\overline{B+C}}$$

【例 1–14】将最简与或式 $Y=AC+BC$ 转换成最简与非－与非式和最简或非－或非式。

解：① 最简与非－与非式。

$$Y=AC+BC=\overline{\overline{AC+BC}}=\overline{\overline{AC}\,\overline{BC}}$$

② 最简或非－或非式。

因为 $Y=AC+BC$，

所以 $\overline{Y}=(\overline{A}+\overline{C})(\overline{B}+\overline{C})=\overline{A}\overline{B}+\overline{C}$。

所以 $Y=\overline{\overline{Y}}=(A+B)C=\overline{\overline{(A+B)C}}=\overline{\overline{(A+B)}+\overline{C}}$。

3. 逻辑函数的代数化简法举例

1）并项法

利用互补律：$A+\overline{A}=1$ 可以将两项合并为一项，并消去一对因子。

【例 1–15】化简函数 $Y=ABC+\overline{AB}C+BD$，写出它的最简与或式。

解：$Y=ABC+\overline{AB}C+BD=(AB+\overline{AB})C+BD=C+BD$。

【例 1–16】试用并项法化简函数 $Y=\overline{A}B\overline{C}+A\overline{C}+\overline{B}\overline{C}$，写出它的最简与或式。

解：$Y=\overline{A}B\overline{C}+A\overline{C}+\overline{B}\overline{C}=\overline{A}B\overline{C}+(A+\overline{B})\overline{C}=\overline{A}B\overline{C}+(\overline{\overline{A}B})\overline{C}=\overline{C}$。

2）吸收法

利用 $AB+\overline{A}C+BC=AB+\overline{A}C$ 和 $A+A\cdot B=A$，将多余项或因子吸收。

【例 1–17】试用吸收法化简函数 $Y=\overline{AB}+\overline{A}C+\overline{B}C$，写出它的最简与或式。

解：$Y=\overline{AB}+\overline{A}C+\overline{B}C=\overline{A}+\overline{B}+\overline{A}C+\overline{B}C=\overline{A}+\overline{B}$。

【例 1–18】试用吸收法化简函数 $Y=AD(B+C+D)$，写出它的最简与或式。

解：$Y=AD(B+C+D)=ADB+ADC+AD=AD+ADC=AD$。

【例 1–19】试用吸收法化简函数 $Y=ABC+\overline{A}D+\overline{C}D+BD$，写出它的最简与或式。

解：$Y=ABC+\overline{A}D+\overline{C}D+BD=ABC+(\overline{A}+\overline{C})D+BD$

$=ABC+\overline{AC}D+BD=ABC+\overline{AC}D=ABC+A\overline{D}+\overline{C}D$。

3）配项法

利用 $A+A=A$，$A=AB+A\overline{B}$ 和 $AB+\overline{A}C=AB+\overline{A}C+BC$ 配项或增加多余项，再和其他项合并。

【例 1–20】试用配项法化简函数 $Y=\overline{A}\overline{B}C+A\overline{B}C+ABC$，写出它的最简与或式。

解：$Y=\overline{A}\overline{B}C+A\overline{B}C+ABC=(\overline{A}\overline{B}C+A\overline{B}C)+(A\overline{B}C+ABC)$

$=\overline{B}C(\overline{A}+A)+AC(\overline{B}+B)=\overline{B}C+AC$。

【例 1–21】试用配项法化简函数 $Y = AB + \overline{A}\overline{B} + \overline{B}C + B\overline{C}$，写出它的最简与或式。

解：$Y = AB + \overline{A}\overline{B} + \overline{B}C + B\overline{C} = AB + \overline{A}\overline{B}(\overline{C} + C) + \overline{B}C + (\overline{A} + A)B\overline{C}$

$= AB + \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{B}C + \overline{A}B\overline{C} + AB\overline{C}$

$= AB(1 + \overline{C}) + \overline{B}C(1 + \overline{A}) + \overline{A}\overline{C}(\overline{B} + B)$

$= AB + \overline{B}C + \overline{A}\overline{C}$。

4）消去法

利用 $A + \overline{A}B = A + B$，$AB + \overline{A}C + BC = AB + \overline{A}C$ 和 $AB + \overline{A}C + BCD = AB + \overline{A}C$ 消去多余项。

【例 1–22】试用消去法化简函数 $Y = AB + \overline{A}C + \overline{B}C$，写出它的最简与或式。

解：$Y = AB + \overline{A}C + \overline{B}C = AB + C(\overline{A} + \overline{B}) = AB + C\overline{AB} = AB + C$。

【例 1–23】试用消去法化简函数 $Y = A\overline{B} + \overline{A}B + ABCD + \overline{A}\overline{B}CD$，写出它的最简与或式。

解：$Y = A\overline{B} + \overline{A}B + ABCD + \overline{A}\overline{B}CD = A\overline{B} + \overline{A}B + CD\overline{A\overline{B} + \overline{A}B} = A\overline{B} + \overline{A}B + CD$。

【例 1–24】试用消去法化简函数 $Y = \overline{AB} + AC + BD$，写出它的最简与或式。

解：$Y = \overline{AB} + AC + BD = \overline{A} + \overline{B} + AC + BD = \overline{A} + \overline{B} + C + D$。

1.7 逻辑函数的卡诺图化简法

1.7.1 逻辑函数的最小项及最小项表达式

1. 最小项的定义

如果 P 是由 n 个变量组成的一个与项，在 P 中每个变量都以原变量或反变量作为一个因子出现一次，且仅出现一次，则称 P 为 n 个变量的一个最小项。显然，n 个变量一共有 2^n 个最小项。

以 3 个变量 A、B、C 为例，共有 $2^3 = 8$ 种取值组合：000，001，010，011，100，101，110 和 111。其对应的与项表示为

$$\overline{A}\overline{B}\overline{C}，\ \overline{A}\overline{B}C，\ \overline{A}B\overline{C}，\ \overline{A}BC，\ A\overline{B}\overline{C}，\ A\overline{B}C，\ AB\overline{C}，\ ABC$$

这 8 个与项的共同特点是：每个与项都有 3 个因子；在每个与项中，A、B、C 每个变量都以原变量或反变量的形式出现，且仅出现一次。

2. 最小项的编号

为了书写方便，对最小项采用编号的形式。编号的方法如下。

（1）将最小项所对应的取值组合看成二进制数，原变量为 1，反变量为 0。

（2）将二进制数转换成十进制数。

（3）该十进制数就是最小项所对应的编号，记作 m_i。

例如，三变量 A、B、C 的一个最小项 $\overline{A}\overline{B}C$ 对应的变量取值组合为 001，将 001 看成二进制数，其所对应的十进制数是 1，即 $(001)_2=(1)_{10}$。所以 $\overline{A}\overline{B}C$ 的编号是 1，记作 m_1。

3. 最小项的性质

（1）任何一个最小项，都对应一组变量的取值组合，有且只有一组变量取值组合使它的值为 1。例如，在三变量 A、B、C 的最小项中，当 $A=1$，$B=0$，$C=1$ 时，$A\overline{B}C=1$。同样的道理，在三变量 A、B、C 的最小项中，当 $A=1$，$B=1$，$C=1$ 时，$ABC=1$。

（2）任何两个最小项的乘积为 0。例如，$\overline{A}\overline{B}\overline{C}\cdot\overline{A}BC=0$。

（3）全部最小项的和为 1。

（4）具有相邻性的两个最小项之和可以合并成一项并消去一对因子。

4. 最小项表达式（标准与或式）

利用真值表可以直接写出逻辑函数的最小项表达式，方法是首先找出使逻辑函数输出值为 1 的所有的输入变量的取值组合，将每组取值组合按照 1 表示原变量、0 表示反变量的方法用与项表示，然后将这些与项进行逻辑加，就得到该逻辑函数的最小项表达式。

【例 1-25】某一含变量 A、B、C 的逻辑函数的真值表如表 1-15 所示，试写出该逻辑函数的标准与或式。

表 1-15　例 1-25 的真值表

输　入			输　出
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

解：根据上述方法，其最小项表达式为

$$Y=\overline{A}\overline{B}C+\overline{A}B\overline{C}+A\overline{B}C+ABC=m_1+m_2+m_5+m_7=\sum m(1,2,5,7)$$。

式中，“$\sum$”为累计或运算符号。

对于任意一个逻辑函数，可以表示成最小项表达式，方法是将每个与项进行等值变

换，将与项中所缺变量补齐。

【例 1–26】将逻辑函数 $Y=AB+BC$ 展开成标准与或式。

解：利用公式 $A+\overline{A}=1$，将所缺变量补齐，则得

$$Y=AB+BC=AB(C+\overline{C})+BC(A+\overline{A})$$

$$=ABC+AB\overline{C}+ABC+\overline{A}BC$$

$$=m_3+m_6+m_7=\sum m(3,6,7)。$$

1.7.2 逻辑函数的卡诺图表示法

卡诺图是由美国工程师莫里斯·卡诺（Maurice Karnaugh）发明的，它比代数法方便、直观、规律性强，可以直接写出函数的“最简”表达式，比较容易掌握，一般运用于五变量以下的函数简化。

1. 卡诺图

卡诺图是逻辑函数的一种图示表示方法。它是将逻辑函数的最小项按一定的规律（相邻性原则）排列成的方格矩阵，每个方格对应一个最小项。如果两个最小项中只有一个变量不同，则称这两个最小项为逻辑相邻，简称相邻项。若将 n 变量的全部 2^n 个最小项用 2^n 个小方格表示，并使具有逻辑相邻性的最小项在几何位置上也相邻地排列起来，则所得的图形称为 n 变量卡诺图。

画变量卡诺图的步骤如下。

（1）根据输入变量的个数确定卡诺图。n 个输入变量的逻辑函数有 2^n 个最小项，因此，应将卡诺图分割成 2^n 个小方格，每个小方格对应一个最小项。

（2）最小项的序号。最小项的序号与方格的序号相同，由方格外边行变量和列变量的取值决定。方格左边是行输入变量，方格上边是列输入变量。如 $A=1$，$BC=01$，对应的小方格序号为 101（或 5），对应的最小项序号为 m_5（或 5）。

（3）变量取值顺序采用的是循环码（格雷码）顺序。变量卡诺图的变量取值之所以按循环码的顺序排列，是为了保证凡是几何相邻的最小项在逻辑上也相邻。下面介绍几何相邻和逻辑相邻的定义和特点。

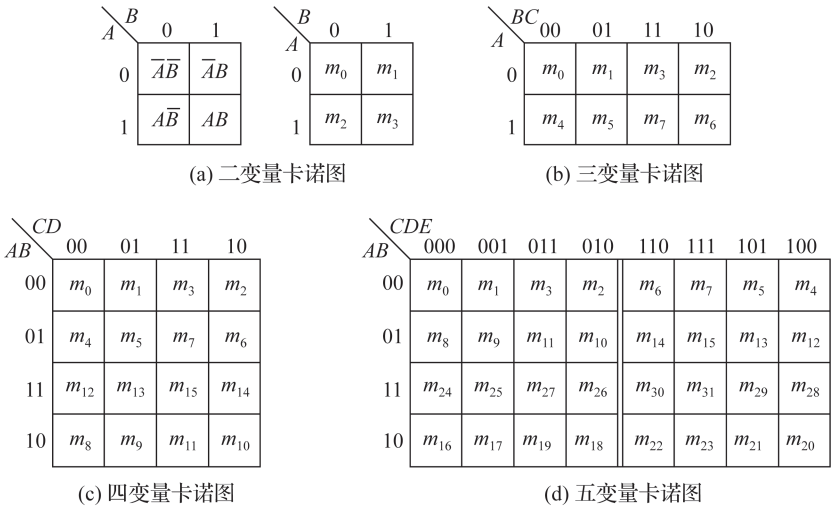
① 几何相邻。最小项在卡诺图中凡是满足下面 3 种情况中 1 种或 1 种以上的就叫几何相邻。

a. 相接：挨着的最小项。

b. 相对：一行或一列两头的最小项。

c. 相重：对折起来能够重合的最小项。

② 逻辑相邻。逻辑相邻是指只有一个变量不同，其余变量都相同的两个最小项在逻辑上是相邻的。例如，$A\overline{B}$ 和 $\overline{A}\overline{B}$ 两个最小项，只有 A 的形式不同，其余变量都相同，所以 $A\overline{B}$ 和 $\overline{A}\overline{B}$ 是逻辑相邻的最小项。图 1–13 是二变量至五变量卡诺图。



(a) 二变量卡诺图　(b) 三变量卡诺图

(c) 四变量卡诺图　(d) 五变量卡诺图

图 1-13　二变量至五变量卡诺图

2. 逻辑函数的卡诺图画法

用卡诺图表示逻辑函数，一般按以下步骤进行。

（1）根据逻辑表达式中的变量 n，绘制 n 变量最小项卡诺图。

（2）在卡诺图上，与逻辑函数中的最小项相对应的位置上填入 1，其余填入 0 或不填。这样就得到了逻辑函数的卡诺图。

【例 1-27】试绘制 $Y=\overline{A}\overline{B}C+\overline{A}B\overline{C}+A\overline{B}C+ABC$ 的卡诺图。

解：$Y=\overline{A}\overline{B}C+\overline{A}B\overline{C}+A\overline{B}C+ABC=m_1+m_2+m_5+m_7=\sum m(1,2,5,7)$。

这是个三变量逻辑函数的最小项表达式。首先绘制一个三变量最小项卡诺图，在卡诺图的 1、2、5 和 7 中分别填入“1”，其余的位置不填，结果如图 1-14 所示。

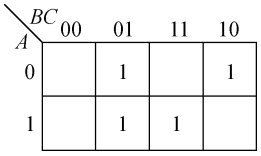


图 1-14　例 1-27 的卡诺图

逻辑函数的卡诺图是逻辑函数的一种表示方法，具有唯一性，即一个逻辑函数只有一个卡诺图。

逻辑函数真值表和逻辑函数的标准与或式是一一对应的关系，所以可以直接根据真值表填卡诺图。

【例 1-28】已知逻辑函数 Y 的真值表如表 1-16 所示，试绘制该逻辑函数 Y 的卡诺图。

表 1-16　例 1-28 的真值表

输　入			输　出
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

解：首先画出三变量卡诺图，如图 1-15 所示。然后将真值表中 $Y=1$ 对应的最小项 m_1、m_2、m_4、m_7 在卡诺图中对应的方格里分别填写 1，其余方格不填。

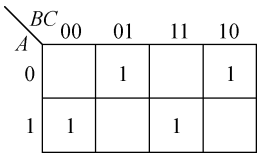


图 1-15　例 1-28 的卡诺图

当一个逻辑函数为一般表达式时，可将其化成标准与或式后再绘制卡诺图。但这样做往往很麻烦，实际上只需把逻辑表达式展开成与或式即可。然后根据与或式每个与项的特征直接填卡诺图。

具体方法是在卡诺图中把每一个与项所含的最小项对应的小方格中均填入 1，直到填入逻辑表达式的全部与项。

【例 1-29】已知逻辑函数 $Y=\overline{A}D+\overline{\overline{AB}(C+\overline{BD})}$，试绘制其卡诺图。

解：（1）把逻辑表达式展开成与或式：

$Y=\overline{A}D+\overline{\overline{AB}(C+\overline{BD})}=\overline{A}D+AB+\overline{C+\overline{BD}}=\overline{A}D+AB+B\overline{C}D$。

（2）绘制四变量最小项卡诺图，如图 1-16 所示。

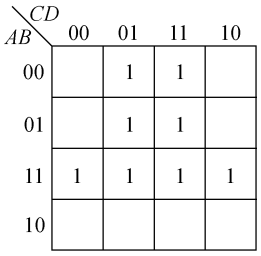


图 1-16　例 1-29 的卡诺图

（3）在与或式每个与项所含的最小项对应的小方格中均填入 1。

第 1 个与项 $\overline{A}D$ 所含最小项中均有 $A=0$ 和 $D=1$，$A=0$ 对应的小方格在第 1 行和第 2 行内；$D=1$ 对应的小方格在第 2 列和第 3 列内，两行和两列相交的小方格为 $\overline{A}D$ 对应的所含最小项。这些小方格的编号为 1、3、5 和 7，故在这 4 个小方格中填入 1。

第 2 个与项是 AB，同理可知，卡诺图中第 3 行所含小方格就是与项 AB，故在编号为 12、13、14、15 的小方格中均填入 1。

同理可知，第 3 个与项 $BC\overline{D}$ 所对应的小方格编号为 5、13，故在这两个小方格中填入 1。

对于有重复最小项的方格只需填入一个 1，如此填入全部与项即可。

1.7.3　用卡诺图化简逻辑函数

卡诺图的相邻性特点保证了几何相邻两方格所代表的最小项只有一个变量不同。因此，多个取值为 1 的方格相邻时，对应的最小项可以合并，消去不同的变量，只保留相同的变量。这就是卡诺图化简法的依据。下面以三变量和四变量卡诺图为例，介绍合并规律。

1. 合并最小项的规律

（1）若两个最小项逻辑相邻，则可合并为一项，同时消去一个互反变量。合并后的结果只剩下公共变量。

例如，图 1–17（a）和图 1–17（b）中画出了 2 个最小项相邻的情况。对于图 1–17（a），m_0 和 m_2 相邻、m_3 和 m_2 相邻、m_5 和 m_7 相邻，所以合并时可以消去一对互反因子。例如，$m_5+m_7=A\overline{B}C+ABC=AC$。

（2）若 4 个最小项逻辑相邻，则可合并为一项，同时消去 2 个互反的变量。合并后的结果只剩下公共变量。

例如，图 1–17（c）和图 1–17（d）的矩形框中为 4 个最小项相邻的情况。对于图 1–17（d）中有 3 组 4 个最小项相邻的情况，它们分别是 m_4、m_5、m_{12} 和 m_{13}，m_3、m_7、m_{15} 和 m_{11}，m_3、m_2、m_{11} 和 m_{10}。第 3 组合并得到

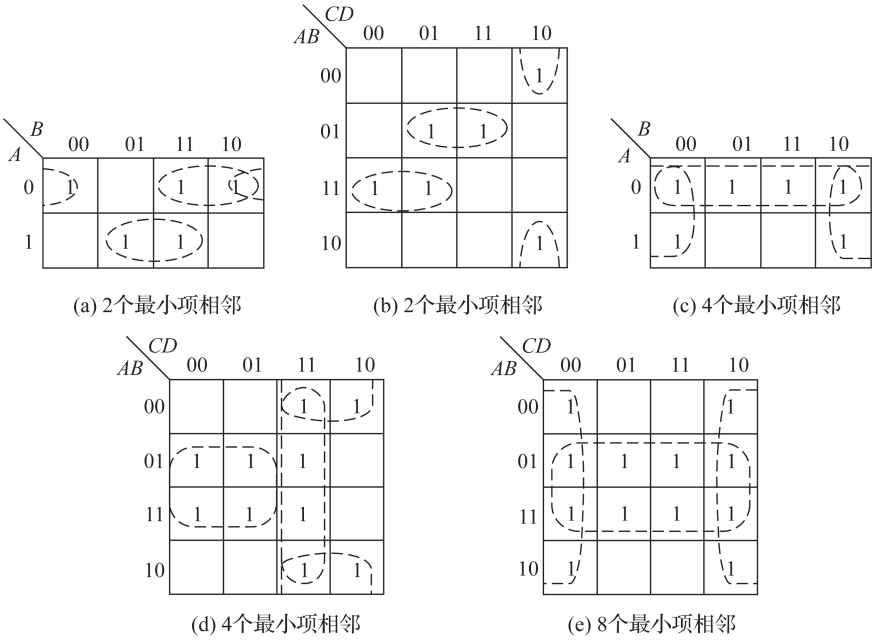
$$\begin{aligned} m_3+m_2+m_{11}+m_{10} &= \overline{A}\overline{B}CD+\overline{A}\overline{B}C\overline{D}+A\overline{B}CD+A\overline{B}C\overline{D} \\ &= \overline{A}\overline{B}C(D+\overline{D})+A\overline{B}C(D+\overline{D}) \\ &= \overline{A}\overline{B}C+A\overline{B}C=(\overline{A}+A)\overline{B}C=\overline{B}C \end{aligned}$$

（3）若 8 个最小项逻辑相邻并且排列成一个矩形组，则可合并为一项并消去 3 个互反变量。合并后的结果只剩下公共变量。

例如，在图 1–17（e）中左右两列的 8 个最小项是相邻的，可将它们合并为一项 $\overline{D}$，其他 3 个变量被消去了。

至此，可以归纳出合并最小项的一般规律：在 n 个变量的卡诺图中，若有 2^k 个方格逻

辑相邻，则它们可以圈在一起加以合并，合并时消去 k 个变量，简化为具有 $n-k$ 个变量的乘积项。若 k 等于 n，则可以消去全部变量，结果为 1。



(a) 2个最小项相邻　(b) 2个最小项相邻　(c) 4个最小项相邻

(d) 4个最小项相邻　(e) 8个最小项相邻

图 1-17　最小项合并时逻辑相邻的几种情况

2. 卡诺图化简逻辑函数的步骤

（1）绘制逻辑函数的卡诺图。

（2）为填“1”的相邻最小项绘制包围圈。

①绘制包围圈的原则。

a. 只有相邻的“1”方格才能合并，且每个包围圈内必须包围 2^n 个相邻的“1”方格。

b. 为了充分简化，“1”可以被重复圈在不同的包围圈中，但新绘制的圈中必须有未被圈过的“1”。

c. 包围圈的个数尽量少，这样逻辑函数的与项就少。

d. 包围圈尽量大，这样消去的变量就多，与门输入端的数目就少。

e. 绘制包围圈时应包含所有的最小项，即覆盖卡诺图中所有的“1”。

②绘制包围圈时应注意的问题。

a. 同一列最上边和最下边循环相邻，可绘制包围圈。

b. 同一行最左边和最右边循环相邻，可绘制包围圈。

c. 4 个角上的“1”方格循环相邻，可绘制包围圈。

（3）分别简化各包围圈。

（4）将各圈简化结果进行逻辑加，得到逻辑函数的最简与或式。

【例 1-30】利用卡诺图化简 $Y=\overline{B}CD+B\overline{C}+\overline{A}\overline{C}D+A\overline{B}C$。

解：第一步，用卡诺图表示该逻辑函数，如图 1–18 所示。

第二步，绘制包围圈。将相邻的“1”方格合并，如图 1–18 所示。

第三步，将所有包围圈最小项的合并结果进行逻辑加，得到逻辑函数的最简与或式。

第一行 2 个“1”方格的包围圈对应的乘积项为 $\overline{A}\overline{B}\overline{C}D+\overline{A}\overline{B}CD=\overline{A}\overline{B}D$。第四行 2 个“1”方格的包围圈对应的乘积项为 $A\overline{B}CD+A\overline{B}C\overline{D}=A\overline{B}C$。中间 4 个“1”方格的包围圈对应的乘积项为 $\overline{A}BC\overline{D}+\overline{A}B\overline{C}D+AB\overline{C}\overline{D}+AB\overline{C}D=B\overline{C}$。将 3 个乘积项求和得到结果，即 $Y=\overline{A}\overline{B}D+A\overline{B}C+B\overline{C}$。

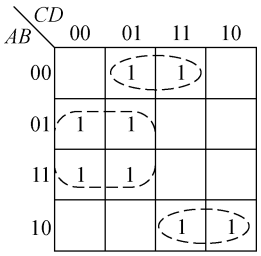


图 1–18　例 1–30 的卡诺图

【例 1–31】用卡诺图简化逻辑函数 $Y=\sum m(0,2,5,7,8,10,12,14,15)$ 为最简与或式。

解：第一步，绘制 4 变量逻辑函数卡诺图，如图 1–19 所示。

第二步，绘制包围圈。将相邻的“1”方格合并，注意卡诺图 4 个角上的“1”方格也是循环相邻的，应圈在一起，故应绘制 4 个包围圈，如图 1–19 所示。

第三步，将所有包围圈最小项的合并结果进行逻辑加，得到逻辑函数的最简与或式为 $Y=B\overline{D}+A\overline{D}+\overline{A}BD+BCD$。

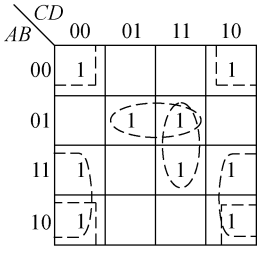


图 1–19　例 1–31 的卡诺图

1.7.4　具有无关项的逻辑函数及其化简

1. 逻辑函数中的无关项

无关项是指那些与所讨论的逻辑问题没有关系的变量取值组合所对应的最小项，这些最小项有两种。

（1）某些变量取值组合不允许出现，如在 8421 码中 1010 ~ 1111 这 6 种代码是不允许出现的，即受到约束，这样的组合称为“约束项”。

（2）某些变量取值组合在客观上不会出现，如在联动互锁开关系统中，几个开关的状态互斥，每次只闭合一个开关。其中一个开关闭合时，其余开关必须断开。因此，在这种系统中两个以上开关同时闭合的情况客观上是不存在的，这样的开关组合称为“随意项”。

约束项和随意项都是一种不会在逻辑函数中出现的最小项，所以对应这些最小项视为1或视为0均可（因为实际上不存在这些变量取值），这样的最小项统称为“无关项”。由无关项加起来所构成的值为0的逻辑表达式称为约束条件。

2. 具有无关项的逻辑函数的化简

在卡诺图中无关项对应的方格常用“×”和“Φ”来标记。在对含有无关项的逻辑函数进行化简时，要充分利用无关项既可看作1也可看作0的特性，尽量扩大卡诺图上所画的圈，才能尽可能多地消除项或变量。

在逻辑表达式中用字母 d 或 Φ 和相应的编号表示无关项。

【例1–32】简化逻辑函数 $Y=\sum m(0,1,4,6,9,13)+\sum d(2,3,5,7,10,11,15)$。式中 $\sum d(2,3,5,7,10,11,15)$ 表示最小项 m_2、m_3、m_5、m_7、m_{10}、m_{11} 和 m_{15} 为无关项。

解：（1）绘制4变量逻辑函数的卡诺图，如图1–20所示。

（2）在逻辑表达式含有的最小项方格中填入1，在无关项方格中填入×。

（3）合并最小项，与“1”方格圈在一起的无关项作为“1”方格，没有圈的无关项丢弃不用（作为0处理）。

（4）写出逻辑函数的最简与或式 $Y=\overline{A}+D$。

显然，利用无关项后的最简与或式更为简单。

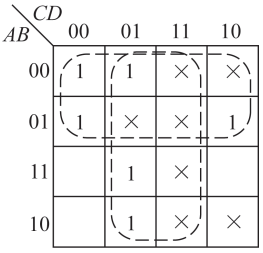


图1–20　例1–32的卡诺图

1.8 正逻辑和负逻辑的规定及其转换

在数字电路中，对逻辑变量的逻辑状态用不同的逻辑体制表示时，所得到的逻辑函数也就不同。正、负逻辑及其真值如表1–17所示，常用正、负逻辑门的逻辑符号如表1–18所示。

在逻辑电路中，若高电平用逻辑1表示，低电平用逻辑0表示，这种逻辑赋值方法称为正逻辑。反之，若高电平用逻辑0表示，低电平用逻辑1表示，这种逻辑赋值方法称

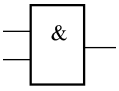
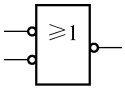
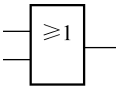
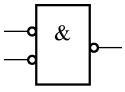
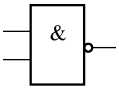
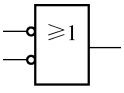
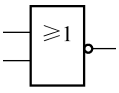
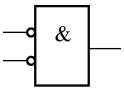
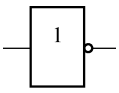
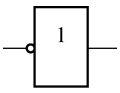
为负逻辑。对于同一电路，采用哪一种逻辑是人为约定的，它并不牵涉电路本身结构的优劣，但是采用的逻辑约定不同，电路所执行的逻辑操作可能完全不同。

一般而言，正、负逻辑的等价关系为正与 = 负或、正或 = 负与、正与非 = 负或非、正或非 = 负与非。

表 1-17　正、负逻辑及其真值表

正逻辑			负逻辑		
A	B	F	A	B	F
0	0	0	1	1	1
0	1	0	1	0	1
1	0	0	0	1	1
1	1	1	0	0	0

表 1-18　常用正、负逻辑门的逻辑符号

正、负逻辑的对偶式	正逻辑的逻辑符号	负逻辑的逻辑符号
$Y = AB = \overline{\overline{A} + \overline{B}}$	正与门： 	负或门： 
$Y = A + B = \overline{\overline{A} \cdot \overline{B}}$	正或门： 	负与门： 
$Y = \overline{A \cdot B} = \overline{A} + \overline{B}$	正与非门： 	负或非门： 
$Y = \overline{A + B} = \overline{A} \cdot \overline{B}$	正或非门： 	负与非门： 
$Y = \overline{A}$	正非门： 	负非门： 

一般情况下，人们习惯于采用正逻辑。但在较复杂的逻辑电路中，有时采用混合逻辑，即正、负逻辑符号同时使用。这时可把整个电路当正逻辑看，而把负逻辑符号中输入端的小圆圈当反向器处理。

逻辑图中任意一条线的两端同时加上或消去小圆圈，其逻辑关系不变。任意一条线一端上的小圆圈移到另一端，其逻辑关系不变。一端消去或加上小圆圈，同时将相应变量取反，其逻辑关系不变。

本章小结

（1）数字信号的数值相对于时间的变化过程是跳变的、间断的。对数字信号进行传输、处理的电子线路称为数字电路。模拟信号通过模 / 数转换后变成数字信号，即可用数字电路进行传输、处理。

（2）数字系统中常用二进制数来表示数据。在二进制位数较多时，也使用十六进制或八进制计数。各种计数体制之间可以相互转换。BCD 码是常用的编码，其中 8421 码使用最广泛。另外，格雷码由于可靠性高，也是一种常用码。

（3）逻辑运算中的三种基本运算是与、或、非。分析数字电路或数字系统的数学工具是逻辑代数。一个逻辑问题可用逻辑函数来描述，逻辑函数有真值表、逻辑表达式、逻辑图等几种常用的表达方法，它们各具特点并可以相互转换。

（4）逻辑函数的公式化简法和卡诺图化简法是本章的重点。公式化简法的优点是没有局限性，但没有固定的模式可以遵循，要求使用者不仅要熟练运用各种公式和定理，还要掌握一定的运算经验和技巧。卡诺图化简法的优点是简单、直观，而且有一定的化简步骤可循，不容易出错，初学者比较容易掌握；但当逻辑变量超过 5 个时，图形复杂，没有实用价值。具有无关项的逻辑函数的化简可以说是逻辑函数化简中的一种特例，也应采用公式法和卡诺图法进行化简。

（5）分析与设计逻辑电路前应搞清楚所采用的逻辑方式是正逻辑还是负逻辑。

课后习题

1. 将下列二进制数转换为十六进制数、八进制数和十进制数。

（1）$(1001101)_2$；

（2）$(11100110)_2$。

2. 比较下列各数，找出最大数和最小数。

$(F8)_{16}$；$(10010101)_2$；$(118)_{10}$。

3. 请将十进制数 75 进行如下变换。

（1）转换为二进制数；

（2）用 8421BCD 码表示；

（3）用 2421 码表示；

（4）用余 3 码表示。

4. 利用公式和定理证明下列等式。

（1）$AB+BCD+\overline{A}C+\overline{B}C=AB+C$；

（2）$AB(C+D)+D+\overline{D}(A+B)(\overline{B}+\overline{C})=A+B\overline{C}+D$。

5. 写出图 1–21 各逻辑图的逻辑表达式，并列出其真值表。

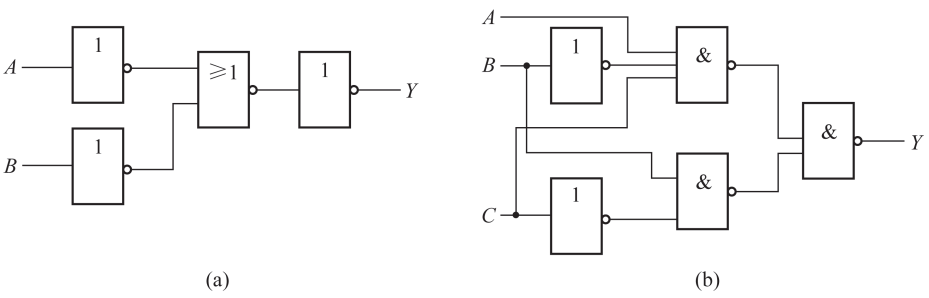


图 1–21　题 5 图

6. 假设图 1–21（a）输入端 A、B 的波形如图 1–22 所示，绘出输出端 Y 的波形。

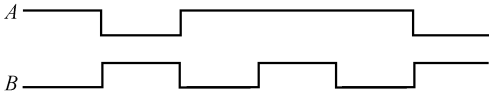


图 1–22　题 6 图

7. 列出逻辑函数 $Y = AB + BC + AC$ 的真值表，并画出逻辑图。

8. 列出下列各逻辑函数的真值表，并说明 Y_1、Y_2、Y_3、Y_4 的关系。

$Y_1 = \overline{A}B + \overline{B}C + \overline{C}A$；　$Y_2 = A\overline{B} + B\overline{C} + C\overline{A}$；

$Y_3 = ABC + \overline{A}\overline{B}\overline{C}$；　$Y_4 = \overline{A\overline{B} + B\overline{C} + C\overline{A}}$。

9. 一个三变量逻辑函数的真值表如表 1–19 所示，写出其最小项表达式，并将其化简为最简与或式。

表 1–19　题 9 真值表

输　入			输　出
A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

10. 将下列各逻辑函数化成最小项之和的形式。

（1）$Y = A + BC + CD$；

（2）$Y=A\overline{B}C+BC+A\overline{C}$；

（3）$Y=(A+B)(\overline{A}+\overline{B}+\overline{C})$。

11. 用代数法将下列逻辑函数化简为最简与或表达式。

（1）$Y=A+A\overline{B}\overline{C}+\overline{A}CD+\overline{C}E+\overline{D}E$；

（2）$Y=AB\overline{C}\overline{D}+A\overline{B}\overline{D}+B\overline{C}D+AB\overline{C}+\overline{B}\overline{D}+B\overline{C}$；

（3）$Y=A+B+C+\overline{A}\overline{B}\overline{C}$；

（4）$Y=(B+\overline{B}C)(A+AD+B)$。

12. 用卡诺图分别将下列函数化为最简与或表达式。

（1）$Y=\overline{(A\oplus B)(C+D)}$；

（2）$Y=A\overline{B}\overline{C}+\overline{A}\overline{B}+\overline{A}D+C+BD$；

（3）$Y=BC+D+\overline{D}(\overline{B}+\overline{C})(AD+B)$；

（4）$Y(A,B,C,D)=\sum m(0,1,3,4,7,12,13,15)$；

（5）$Y(A,B,C,D)=\sum m(0,1,2,5,8,9,10,12,14)$。

13. 用卡诺图分别将下列含有无关项的函数化为最简与或表达式。

（1）$Y(A,B,C,D)=\sum m(0,2,4,9)+\sum d(10,11,12,13,14,15)$；

（2）$Y(A,B,C,D)=\sum m(0,2,7,13,15)+\sum d(1,3,4,5,6,8,10)$；

（3）$Y(A,B,C,D)=\sum m(3,5,6,7,10)$ [约束条件：$\sum d(0,1,2,4,8)=0$]；

（4）$Y=C\overline{D}(A\oplus B)+\overline{A}B\overline{C}+\overline{A}C\overline{D}$（约束条件：$AB+CD=0$）。