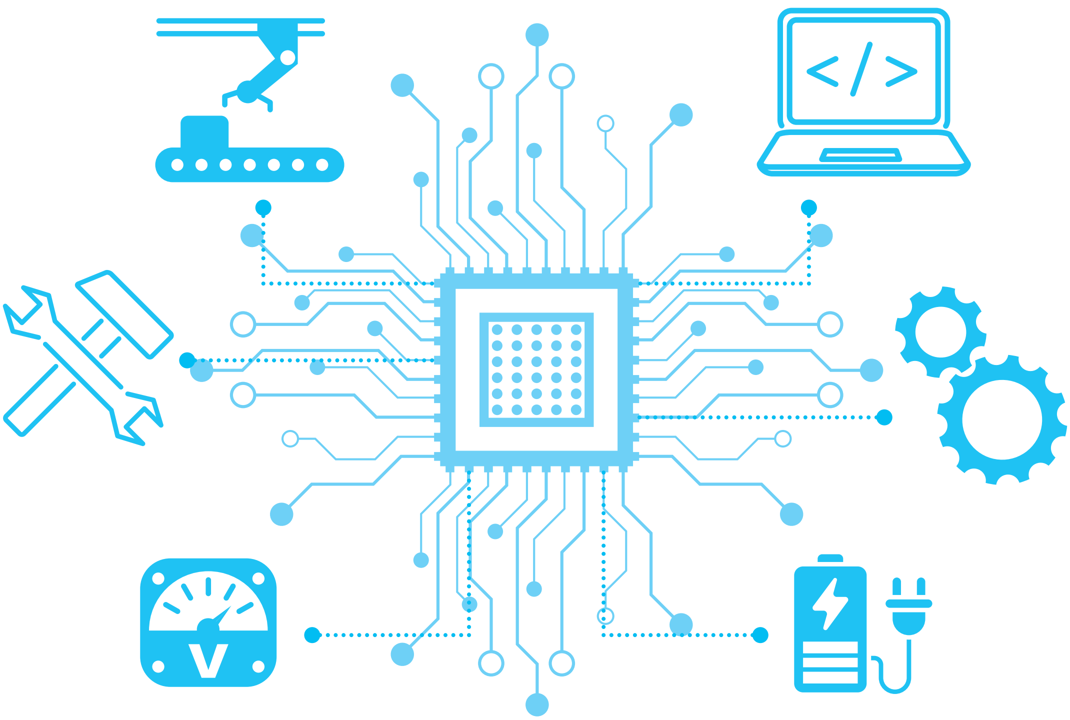
高等职业教育机电系列精品教材

数字电子技术

主　编　韩睿群　王晓岚　刘　越

副主编　张文静

主　审　郑勇峰



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

内容提要

本书全面系统地介绍了数字电子技术的基础知识。全书共分为8章,包括数字电路基础知识、逻辑门电路、组合逻辑电路、触发器、时序逻辑电路、存储器和可编程逻辑器件、脉冲波形的产生与整形、数模和模数转换。

本书既可作为高等职业院校电子信息工程、通信工程、计算机科学与技术等相关专业的教材,也可作为相关领域工程技术人员的参考书。

图书在版编目(CIP)数据

数字电子技术 / 韩睿群，王晓岚，刘越主编. 上海：上海交通大学出版社，2025. 8. -- ISBN 978-7-313-33196-0

Ⅰ. TN79

中国国家版本馆 CIP 数据核字第 2025A7265Y 号

数字电子技术

SHUZI DIANZI JISHU

主　　编:韩睿群　王晓岚　刘　越

出版发行:上海交通大学出版社　　地　　址:上海市番禺路951号

邮政编码:200030　　电　　话:021-64071208

印　　制:大厂回族自治县聚鑫印刷有限责任公司　　经　　销:全国新华书店

开　　本:787 mm×1 092 mm　1/16　　印　　张:12

字　　数:233千字

版　　次:2025年8月第1版　　印　　次:2025年8月第1次印刷

书　　号:ISBN 978-7-313-33196-0　　电子书号:ISBN 978-7-89564-399-4

定　　价:45.00元

前言

随着信息技术的飞速发展，数字电子技术作为现代信息技术的基础和核心，已成为推动社会进步和产业升级的重要力量。无论是通信、计算机、自动化控制、电子仪器仪表等应用领域，还是人工智能、物联网等新兴领域，都离不开数字电子技术的支撑。因此，培养具备扎实理论基础与较强实践能力的数字电子技术人才，是目前高等职业教育的重要任务之一。基于这一背景，编者编写了本书，旨在为学生提供一本既反映学科前沿动态，又契合教学实际需求的数字电子技术教材。

本书全面系统地介绍了数字电子技术的基本理论、基本概念、基本分析方法及典型应用。内容涵盖数字电路基础知识、逻辑门电路、组合逻辑电路、触发器、时序逻辑电路、存储器和可编程逻辑器件、脉冲波形的产生与整形、数模和模数转换等。此外，特别强调了数字电路的设计与仿真方法，以及这些方法在真实案例中的应用，力求使学生不仅能掌握理论知识，还能将其灵活应用于实践。

本书注重培养学生的创新思维和实践能力，通过设置综合性、设计性实验，鼓励学生进行独立思考和动手实践。同时，引导学生参与科研创新活动，应用所学知识解决实际问题，提升学生的综合素质和创新能力。本书介绍了数字电子技术领域的最新成果和技术动态，保持了教材的先进性和时效性。本书章节安排合理，逻辑性强，便于学生系统地学习和掌握知识。丰富的配套资源（如习题及答案、课后实验、多媒体教学课件等）能充分满足不同层次学生的学习需求。

本书由长期从事数字电子技术教学与科研工作的一线骨干教师领衔编写，团队成员具有丰富的教学经验和深厚的学术造诣。编者结合多年的教学实践和科研成果，精心编写了本书，力求做到科学严谨、内容全面、实用性强。

本书由天津渤海职业技术学院韩睿群、王晓岚、刘越任主编，张文静任副主编。具体编写分工如下：韩睿群编写第 3 章、第 4 章、第 8 章；王晓岚编写第 5 章至 7 章及实验内容；刘越编写第 2 章；张文静编写第 1 章。本书由天津渤海职业技术学院郑勇峰教授担任主审。

在本书编写的过程中，编者参考了大量的文献，在此对文献的作者表示衷心的感谢！

由于编者水平有限，书中难免有不当之处，恳请广大读者批评指正。

编　者

2025 年 1 月

目录

第 1 章　数字电路基础知识

课前导入

电子电路按其处理信号的不同通常可分为模拟电子电路及数字电子电路两大类，分别简称模拟电路及数字电路。

模拟电路处理的信号是模拟信号。模拟信号在时间和幅值上都是连续变化的。如温度、压力等实际的物理信号，各种温度及压力检测仪表输出的模拟温度、压力变化的电信号，模拟语音的音频电信号等。

数字电路处理的信号是数字信号。数字信号在时间和幅值上都是离散的。如刻度尺的读数、数字显示仪表的显示值以及各种门电路的输入输出信号等。

数字电路是构成各种数字系统尤其是数字电子计算机的基础。

学习目标

(1)掌握数制和编码的概念。

(2)掌握逻辑代数的基本概念和逻辑运算。

(3)理解数字电路的基本概念，了解数字信号、二进制数及数字电路的重要性。

技能目标

(1)掌握逻辑函数的化简方法。

(2)能理解和表达逻辑函数。掌握逻辑函数的表示方式，如逻辑表达式、真值表、逻辑电路图等。

素质目标

(1)养成自觉遵守规则、诚实守信的良好习惯。

(2)培养正确的价值观。

(3)激发求知欲和拼搏精神，为学业和人生道路提供动力。

知识准备

1.1 数制与编码

1.1.1 数制

数字电路经常会遇到计数的问题。计数时，如果一位数不够用就要用多位数进行表示。人们把多位数中的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。在日常生活中，最常见的数制是十进制，而数字电路在进行数字运算和处理时，广泛采用的数制是二进制。但二进制数有时表示起来不太方便，位数太多，所以实际应用中也常采用八进制和十六进制。

视频

数制与码制

1. 几种常见数制的表示方式

1）十进制

十进制是用 10 个不同的数字符号 0、1、2、3、4、5、6、7、8、9 来表示十进制数的，所以其计数的基数是 10。超过 9 的十进制数必须用多位数表示，其中低位数和相邻的高位数之间的关系是“逢十进一”，故称十进制。例如：

视频

十进制

$$314.25=3\times10^2+1\times10^1+4\times10^0+2\times10^{-1}+5\times10^{-2}$$

等号右边的表示方式称为十进制数的多项式表示法，也叫按权展开式。数字符号所处的位置不同，所代表的数值也不同，即权值不同。例如，3 处在百位，代表 300，即 3×100，也可以说 3 的权值是 10^2。由此可知，上式各位的权值分别为 10^2、10^1、10^0、10^{-1}、10^{-2}。

实际上，任意一个十进制数 N 都可以用多项式表示法写成如下形式：

$$\begin{aligned}(N)_{10} &=\pm(K_{n-1}\times10^{n-1}+K_{n-2}\times10^{n-2}+\cdots+K_1\times10^1+K_0\times10^0+\\&\qquad K_{-1}\times10^{-1}+\cdots+K_{-m}\times10^{-m})\\&=\sum_{i=-m}^{n-1}K_i\times10^i \qquad (1.1)\end{aligned}$$

式中，K_i 表示第 i 位的数字符号（以下简称数符），它可以是 0、1、2、3、4、5、6、7、8、9 中的任意一个，由具体的十进制数 N 来确定；10^i 为第 i 位的权值；i 为数符 K_i 的位置号，小数点前的第一位 $i=0$，第二位 $i=1$，以此类推；小数点后的第一位 $i=-1$，第二位 $i=-2$，以此类推；n，m 为正整数，n 表示整数位数，m 表示小数位数。

任意一个十进制数如 313.25,可以书写成 313.25、$(313.25)_{10}$ 或 313.25D(D 表示十进制)的形式。

2)二进制

二进制的基数为 2,即它使用的数符只有 0、1 两个,它的进位规则是“逢二进一”。任意一个二进制数 N 都可以用多项式表示法写成如下形式:

视频

二进制

$$(N)_2 = \sum_{i=-m}^{n-1} K_i \times 2^i \tag{1.2}$$

式中,K_i 表示二进制数第 i 位的数字符号,它可以是 0 或 1,由具体的二进制数 N 来确定;m、n 为正整数,n 表示整数位数,m 表示小数位数。例如:

$$(11011.101)_2 = 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$$

其代表十进制数 27.625。

任意一个二进制数如 11011.101,可以书写成$(11011.101)_2$或 11011.101B(B 表示二进制)的形式。

二进制的优点是它只有两个数符,因此这两个数符可以用任何具有两个不同稳定状态的元器件来表示,如三极管的饱和与截止、继电器的闭合与断开、灯泡的亮与灭等。只要规定其中一种状态为 1,另一种状态就是 0。多个元器件的不同状态组合就可以表示一个二进制数,因此二进制数的存储、传送过程简单可靠。在数字系统和计算机内部,数据的表示与存储都是以二进制形式进行的。很显然,十进制的数符需要采用具有 10 个稳定状态的元器件来表示,这不仅给技术的实现带来许多困难,而且不经济。此外,二进制还有运算规律简单的优点,这使采用二进制的电路的运算控制更加简单。

当然二进制也有缺点。用二进制表示一个数时,采用的位数过多。例如,十进制数 49,表示成二进制数为 110001。这种方式使用起来既不方便,也不符合人们的日常习惯。为了便于读写,通常有两种解决办法。一种是原始数据还采用十进制,在将其输入数字系统时,将原始数据转换成数字系统能接受的二进制数,而在数据运算处理结束后,再将二进制数转换成十进制数输出,以表示最终结果。另一种办法是采用八进制或十六进制。

3)八进制

八进制的基数为 8,即它使用的数字符号只有八个,它们是 0、1、2、3、4、5、6、7,它的进位规则是“逢八进一”。任意一个八进制数 N 都可以用多项式表示法写成如下形式:

视频

八进制、十六进制

$$(N)_8 = \sum_{i=-m}^{n-1} K_i \times 8^i \tag{1.3}$$

式中,K_i 表示第 i 位的数符,它可以是 0、1、2、3、4、5、6、7 中的任意一个,由具体的八进制数 N 来确定;每位数符的权值是基数 8 的不同次幂;m、n 表示正整数,m 表

示小数位数,n 为整数位数。例如:

$$(61)_8 = 6\times8^1+1\times8^0$$

其代表十进制数 49。

任意一个八进制数如 61,可以书写成$(61)_8$或 61O(O 表示八进制)的形式。

4)十六进制

十六进制的基数为 16,即它使用的数符有 16 个,它们是 0、1、2、3、4、5、6、7、8、9、A(10)、B(11)、C(12)、D(13)、E(14)、F(15)。它的进位规则是“逢十六进一”。任意一个十六进制数 N 都可以用多项式表示法写成如下形式:

$$(N)_{16} = \sum_{i=-m}^{n-1} K_i \times 16^i \tag{1.4}$$

式中,K_i 表示第 i 位的数符,它可以是 0~F 中的任意一个,由具体的十六进制数 N 来确定;每位数符的权值是基数 16 的不同次幂;m、n 表示正整数,m 表示小数位数,n 为整数位数。例如:

$$(1A5)_{16}=1\times16^2+A\times16^1+5\times16^0$$

其代表十进制数 421。

任意一个十六进制数如 31,可以写成$(31)_{16}$或 31H(H 表示十六进制)的形式。

几种数制的对应关系如表 1-1 所示。

表 1-1　几种数制的对应关系

十 进 制	二 进 制	八 进 制	十 六 进 制
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

2. 数制转换

同一个数可以用不同的数制表示，例如，十进制数 49，表示成二进制数是 110001B，表示成八进制是 61O，表示成十六进制是 31H。一个数从一种数制表示变成另一种数制表示，称为数制转换。下面介绍数制转换的方法。

视频

十进制与二进制的相互转换

1）二进制数、八进制数、十六进制数转换为十进制数

二进制数、八进制数、十六进制数转换为十进制数可以采用多项式表示法。具体方法：将二进制数（或八进制数，或十六进制数）用多项式表示法写出，然后按十进制运算规则算出相应的十进制数值即可。现举例说明。

［例 **1.1.1**］ 将二进制数 11011.101 转换为十进制数。

［解］ $(11011.101)_2=1\times2^4+1\times2^3+0\times2^2+1\times2^1+1\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}$

$=16+8+0+2+1+0.5+0+0.125=(27.625)_{10}$

［例 **1.1.2**］ 将八进制数 35 转换为十进制数。

［解］ $(35)_8=3\times8^1+5\times8^0=(29)_{10}$

［例 **1.1.3**］ 将十六进制数 3F5 转换为十进制数。

［解］ $(3F5)_{16}=3\times16^2+15\times16^1+5\times16^0=(1013)_{10}$

2）十进制数转换为二进制数、八进制数、十六进制数

（1）十进制整数转换为二进制、八进制、十六进制整数。十进制整数转换为任意数制整数的方法是“除基取余”，具体过程：除基取余，直至商为 0，余数的高位到低位的排列顺序为由下到上。

［例 **1.1.4**］ 将十进制数 58 转换为二进制数。

［解］ 整个推算过程如下：

2	58	…………………	余数＝0，$B_0=0$
2	29	…………………	余数＝1，$B_1=1$
2	14	…………………	余数＝0，$B_2=0$
2	7	…………………	余数＝1，$B_3=1$
2	3	…………………	余数＝1，$B_4=1$
2	1	…………………	余数＝1，$B_5=1$
	0		

转换结果为 $(58)_{10}=B_5B_4B_3B_2B_1B_0=(111010)_2$。

［例 1.1.5］ 将十进制数 278 转换为八进制数。

［解］

8	278	…………………	余数=6，$H_0=6$
8	34	…………………	余数=2，$H_1=2$
8	4	…………………	余数=4，$H_2=4$
	0		

转换结果为$(278)_{10}=H_2H_1H_0=(426)_8$。

［例 1.1.6］ 将十进制数 2 803 转换为十六进制数。

［解］

16	2 803	…………………	余数=3，$H_0=3$
16	175	…………………	余数=15，$H_1=\mathrm{F}$
16	10	…………………	余数=10，$H_2=\mathrm{A}$
	0		

转换结果为$(2\,803)_{10}=H_2H_1H_0=(\mathrm{AF3})_{16}$。

(2)十进制小数转换为二进制小数。十进制小数转换为二进制小数的方法为“乘 2 取整”，具体方法：乘基取整，取有效位，小数点后的数的高位到低位排列顺序为由上到下。

［例 1.1.7］ 将十进制小数 0.562 5 转换成二进制小数，过程如下。

［解］

0.562 5	
× 2	
1.125 0	整数部分=1，$B_{-1}=1$
× 2	
0.250 0	整数部分=0，$B_{-2}=0$
× 2	
0.500 0	整数部分=0，$B_{-3}=0$
× 2	
1.000 0	整数部分=1，$B_{-4}=1$

转换结果为$(0.562\,5)_{10}=0.B_{-1}B_{-2}B_{-3}B_{-4}=(0.1001)_2$。

实际应用中，一般根据精度要求，只取有限位即可。

此外，如果一个数既有整数部分，也有小数部分，那么可用前述的“除基取余”及“乘基取整”的方法分别对整数部分和小数部分进行转换，然后将转换结果合并起来即可。例如：

$$\begin{array}{ccc} 17.25 = & 17 & + \; 0.25 \\ & \downarrow & \downarrow \\ & 10001 & 0.01 \end{array}$$

转换结果为$(17.25)_{10}=(10001.01)_2$。

3)八进制数与二进制数的相互转换

由于三位二进制数刚好有8种不同的数位组合，如下所示。

二进制：000，001，010，011，100，101，110，111。

八进制：0，1，2，3，4，5，6，7。

因此可以用三位二进制数表示一位八进制数，而一位八进制数也可以用三位二进制数表示。

视频

二、八、十六进制相互转换

[**例1.1.8**] 将八进制数53.21转换成二进制数。

[**解**]

八进制：	5	3.	2	1
	↓	↓	↓	↓
二进制：	101	011.	010	001

转换结果为$(53.21)_8=(101011.010001)_2$。

[**例1.1.9**] 将二进制数101111.011转换成八进制数。

[**解**]

二进制：	101	111.	011
	↓	↓	↓
八进制：	5	7.	3

转换结果为$(101111.011)_2=(57.3)_8$。

4)十六进制数与二进制数之间的相互转换

由于四位二进制数0000～1111的16种组合刚好可以代表十六进制数的16个数符，因此可用四位二进制数表示一位十六进制数，而一位十六进制数也可以用四位二进制数表示。十六进制数与二进制数的转换方法同八进制数与二进制数的转换方法类似。

[**例1.1.10**] 将十六进制数D3F5转换成二进制数。

[**解**]

十六进制：	D	3	F	5
	↓	↓	↓	↓
二进制：	1101	0011	1111	0101

转换结果为$(D3F5)_{16}=(1101001111110101)_2$。

[**例1.1.11**] 将二进制数101111转换成十六进制数。

[**解**]

二进制：	0010	1111
	↓	↓
十六进制：	2	F

转换结果为$(101111)_2=(2F)_{16}$。

显然，八进制或十六进制比二进制书写更简短、易读，便于记忆，而且其与二进制数的转

换非常方便，因此在数字系统和计算机中，原始数据经常采用八进制数或十六进制数书写，而在数字系统和计算机内部，数据则是用二进制数表示的。

1.1.2 编码

在数字系统和计算机内部，数据是以二进制形式存在的，那么对于各种字符，如字母、标点符号及汉字等信息，机器是如何进行识别和处理的呢？这就涉及编码的问题了。将若干位二进制数按一定的组合方式组合起来以表示数（包括大小和符号）和字符等信息的过程，就叫编码。编码方式即码制有多种，下面介绍常用的编码方式。

人们最熟悉、习惯的计数方式是十进制计数方式，而在数字系统和计算机中数据只能用二进制表示，但用二进制表示十进制数的位数过多，不便于读写。为了解决这一问题，可以把十进制数的每位数字用若干位二进制数符表示。通常将这种用若干位二进制数符表示一位十进制数的编码方法称为二进制编码的十进制，简称 BCD(binary coded decimal)码。

常见的 BCD 码有 8421 码、2421 码等。其中 8421 码是最基本、最常见的一种 BCD 码。它是将十进制数的每个数符用四位二进制数码来表示的一种编码方式。该四位二进制数的每位都有固定的权值，从左至右各位的权值分别为 8、4、2、1，故称 8421 码。表 1-2 列出了十进制数与 8421 码的对应关系。

表 1-2　十进制数与 8421 码的对应关系

十进制数	8421 码
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

从表 1-2 可以看出，每个十进制数符所对应的二进制编码就是与该十进制数符等值的二进制数。因此在 8421 码中不可能出现 1010、1011、1100、1101、1110、1111 等编码形式。

任何一个十进制数采用 8421 码表示时，只要把该十进制数的每位转换成相应的 8421 码即可。例如：

$$(129)_{10}=(0001\ 0010\ 1001)_{8421}$$

同样，任何一个 8421 码表示的十进制数，也可以方便地转换成普通十进制数。例如：

$$(0101\ 0111\ 1001\ 0001)_{8421}=(5791)_{10}$$

采用 8421 码表示的十进制数与采用二进制数表示的十进制数相比，转换更加方便、直观。

1.2　逻辑代数基础

逻辑代数又叫布尔代数或开关代数，是英国数学家乔治・布尔（George Boole）在 19 世纪中期创立的。它是分析设计数字电路的基础。与普通代数不同，逻辑代数研究的是逻辑变量与逻辑函数之间的关系。在逻辑代数中，有些运算规则与普通代数的运算规则相同，有些则完全不同，在学习逻辑代数的过程中，应加以注意。

1.2.1　逻辑变量与逻辑函数

1. 逻辑变量

虽然逻辑代数和普通代数一样，也是用字母表示变量的，但它的变量是逻辑变量，只取两个值，0 或 1，没有中间值。而且，这里的 0 或 1 并不代表数量的大小，而是代表两种不同的逻辑状态，如是与非、真与假、高与低、有与无、闭合与断开等。

2. 逻辑函数

对于表达式 $Y=AB$，也可以记为 $Y=F(A,B)=AB$，即认为 Y 是 A、B 的函数。由于 A、B 都是逻辑变量，所以 Y 也是逻辑变量。Y 与 A、B 的关系是逻辑“与”关系。这样的函数 $F(A,B)$ 就称为逻辑函数。

一般情况下，逻辑函数可记为 $F(A_1,A_2,\cdots,A_n)$，其中 $A_1,A_2,\cdots,A_n$ 是逻辑自变量，如 $F(A,B,C)=AB+\overline{A}C+B\overline{C}$。逻辑函数与普通函数相比有以下两个特点。

（1）逻辑函数中的变量只有 0 或 1 两种取值。

（2）逻辑函数中的变量之间的运算关系只能是“与”“或”“非”三种基本逻辑关系。

1.2.2　逻辑代数的基本逻辑运算及运算定律

像普通代数有自己的基本运算及运算定律一样，逻辑代数也有自己的基本运算及运算定律。

视频

逻辑代数的基本运算

1. 基本逻辑运算

1）“与”运算

图 1-1 所示为“与”电路。

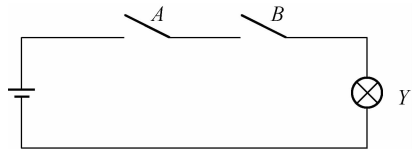


图 1-1 “与”电 路

如果将开关闭合记为 1，开关断开记为 0，灯亮记为 1，灯灭记为 0，那么该电路只有在两个开关都闭合时，灯才亮。用 A 和 B 表示两个逻辑变量，用 Y 表示结果，如果将 A、B 所有可能的取值及其进行“与”运算的全部可能结果 Y 列成表，如表 1-3 所示，那么这样的表称为真值表。

表 1-3 “与”运算真值表

A	**B**	**Y=AB**
0	0	0
0	1	0
1	0	0
1	1	1

从表 1-3 中可以看出，只有当 A 与 B 都为 1 时，Y 才为 1。因此对“与”电路工作状态的描述就可以用“与”运算表达：

$$Y=AB$$

其运算规律为“有 0 为 0，全 1 为 1”，读作“Y 等于 A 乘 B”，或“Y 等于 A 与 B”。这里，A、B、Y 都是逻辑变量，A、B 代表进行“与”运算的两个变量，Y 是运算结果。

2)“或”运算

在图 1-2 所示的电路中，如果将开关闭合记为 1，开关断开记为 0，灯亮记为 1，灯灭记为 0，那么该电路只要有一个开关闭合，灯就亮。

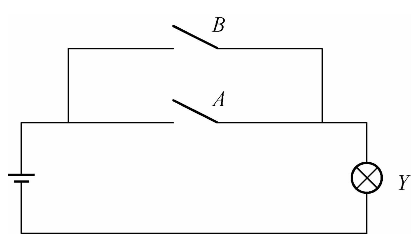


图 1-2 “或”电 路

用 A 和 B 表示两个逻辑变量，用 Y 表示结果。将 A 和 B 的全部可能取值及运算的全部可能结果 Y 列成真值表，如表 1-4 所示。

表 1-4 “或”运算真值表

A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

从表 1-4 可以看出，只要 A 或 B 有一个为 1，Y 就为 1。对该电路工作状态的描述可以用”或”运算表达：

$$Y=A+B$$

其运算规律为：“有 1 为 1，全 0 为 0”，读作“Y 等于 A 加 B”，或“Y 等于 A 或 B”。

3)“非”运算

“非”运算也叫逻辑反或逻辑否定，其运算符记为“－”。“非”运算的逻辑表达式为

$$Y=\overline{A}$$

读作“Y 等于 A 非”，或“Y 等于 A 反”。这里，A 代表进行”非”运算的逻辑变量，Y 是运算结果。“非”运算的真值表如表 1-5 所示。

表 1-5 “非”运算真值表

A	$Y=\overline{A}$
1	0
0	1

“非”运算的特点是若 A 为 1，则 $\overline{A}$ 为 0；若 A 为 0，则 $\overline{A}$ 为 1。

2. 基本运算定律

根据“与”“或”“非”逻辑运算的基本法则，可推导出逻辑代数的基本运算定律，如表 1-6 所示。

表 1-6 逻辑代数的基本运算定律

	加	乘	非
基本定律	$A+0=A$	$A\cdot 0=0$	$A+\overline{A}=1$
	$A+1=1$	$A\cdot 1=A$	$A\cdot\overline{A}=0$
	$A+A=A$	$A\cdot A=A$	$\overline{\overline{A}}=A$
	$A+\overline{A}=1$	$A\cdot\overline{A}=0$	

续表

结　合　律	$(A+B)+C=A+(B+C)$
	$(AB)C=A(BC)$
交　换　律	$A+B=B+A$
	$AB=BA$
分　配　律	$A(B+C)=AB+AC$
	$A+BC=(A+B)(A+C)$
摩根定律(反演律)	$\overline{ABC\cdots}=\overline{A}+\overline{B}+\overline{C}+\cdots$
	$\overline{A+B+C+\cdots}=\overline{A}\,\overline{B}\,\overline{C}\cdots$
吸　收　律	$A+AB=A$
	$A(A+B)=A$
	$A+\overline{A}B=A+B$
	$A(\overline{A}+B)=AB$
	$(A+B)(A+C)=A+BC$
包　含　律	$AB+\overline{A}C+BC=AB+\overline{A}C$
	$AB+\overline{A}C+BCD=AB+\overline{A}C$

上述每一个公式都可以方便地用真值表进行直接证明。

[**例 1.2.1**]　证明摩根定律(反演律)。

[**解**]　为简单起见,以证明公式$\overline{A+B}=\overline{A}\,\overline{B}$ 为例。列出其真值表,如表 1-7 所示。

表 1-7　例 1.2.1 的真值表

A	B	$\overline{A+B}$	$\overline{A}\,\overline{B}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

由真值表可知,摩根定律公式成立。

采用真值表的方法对基本运算定律进行证明时,如果自变量较多,那么这种方法是比较麻烦的,但其优点是可以直接证明,不依赖其他定律。

在证明其他逻辑表达式或进行逻辑函数化简时,可直接利用上面给出的基本运算定律。

视频

公式及运算法则

1.2.3　逻辑代数的重要准则

1. 代入准则

任何一个含有某变量 A 的等式,如果将所有出现 A 的位置都用一

个逻辑函数 F 代换，则等式仍然成立，这个准则称为代入准则。

因为任何一个逻辑函数和逻辑变量一样，只有 0 和 1 两种可能的取值，所以代入准则是成立的。

例如，在 $B(A+C)=BA+BC$ 中，将所有出现 A 的地方都用函数 $A+D$ 代换，则等式仍成立，即得：

$$B[(A+D)+C]=B(A+D)+BC=BA+BD+BC$$

再如，$AB+A\overline{B}=A$，将 B 用函数 CD 代换，则等式 $ACD+A\overline{CD}=A$ 仍成立。

值得注意的是，在使用代入准则时，一定要把等式中所有需要代换的变量全部代换掉，否则代换后所得的等式将不成立。

2. 反演准则

设 F 是一个逻辑函数表达式，如果将 F 中所有的“与”运算符变为“或”运算符，“或”运算符变为“与”运算符；原变量变为反变量，反变量变为原变量，所得到的新的逻辑函数表达式就是 $\overline{F}$，这就是反演准则。

反演准则是反演律的推广。利用反演准则可以很容易地求出函数的“反”。

［例 1.2.2］ 已知 $Y=AB+\overline{C}D$，求 $\overline{Y}$。

［解］ 根据反演准则可得：$\overline{Y}=(\overline{A}+\overline{B})(C+\overline{D})$。

［例 1.2.3］ 已知 $Y=A[\overline{B}+(C\overline{D}+\overline{E}F)]$，求 $\overline{Y}$。

［解］ 根据反演准则可得：$\overline{Y}=\overline{A}+B(\overline{C}+D)(E+\overline{F})$。

3. 对偶准则

设 F 是一个逻辑函数表达式，如果将 F 中所有的“与”运算符变为“或”运算符，“或”运算符变为“与”运算符，所得到的新的逻辑函数表达式就是 F 的对偶式，记作 F'，这就是对偶准则。当某个逻辑恒等式成立时，其对偶式也成立。

通过前面的逻辑代数的基本运算定律表，不难看出，这些公式总是成对出现的，例如：

$A+B=B+A$ 和 $AB=BA$（交换律）。

$A(B+C)=AB+AC$ 和 $A+BC=(A+B)(A+C)$（分配律）等。

这些式子都互为对偶式。

1.2.4 逻辑函数的公式化简法

视频

公式化简

一个逻辑函数可以有多种不同的表达式，例如，$Y=(A+B)(A+C)$，也可以写成 $Y=A+BC$。如果对逻辑函数进行分类，其主要有“与或”表达式，“或与”表达式，“与非与非”表达式，“或非或非”表达式以及

“与或非”表达式等形式。例如：

$$
\begin{aligned}
Y &= AB+\overline{A}C && \text{“与或“表达式} \\
&= (A+C)(\overline{A}+B) && \text{“或与”表达式} \\
&= \overline{\overline{AB}\cdot\overline{\overline{A}C}} && \text{“与非与非”表达式} \\
&= \overline{\overline{A+C}+\overline{\overline{A}+B}} && \text{“或非或非”表达式} \\
&= \overline{A\,\overline{B}+\overline{A}\,\overline{C}} && \text{“与或非”表达式}
\end{aligned}
$$

即使同一种类型的表达式，其形式也不是唯一的。如上面的“与或”表达式：

$$F=AB+\overline{A}C=AB+\overline{A}C+BC=ABC+AB\,\overline{C}+\overline{A}BC+\overline{A}\,\overline{B}C=\cdots$$

由于表达式的繁简程度不同，所以实现它们的逻辑电路(第 2 章将做具体介绍)也不相同。一般来说，如果表达式比较简单，那么实现它们的逻辑电路所需要使用的元器件就比较少，电路结构也比较简单。

那么什么样的表达式是最简的？我们以“与或”表达式为例。所谓最简的“与或”表达式，通常是指：

(1)表达式中的乘积项(或项)的个数最少。

(2)在满足(1)的前提下，每个乘积项中变量的个数最少。

只要得到了最简“与或”表达式，就不难得到其他类型的最简表达式。

为了简化逻辑电路，应先得到最简表达式，所以就需要对逻辑函数进行化简。常用的化简方法有公式化简法(代数化简法)和卡诺图法(图解法)。

公式化简法是运用逻辑代数的基本定律和准则对逻辑函数进行化简的方法。由于逻辑函数的表达式是多种多样的，公式化简法还未形成一套完整的方法。能否以最快的速度进行化简以得到最简表达式？这与我们的经验和对公式掌握与运用的熟练程度有密切的关系。

[**例 1.2.4**]　化简函数 $F=A\,\overline{B}C+\overline{AB}\,\overline{C}$。

[**解**]　由分配律可得：

$$
\begin{aligned}
F &= A\,\overline{B}C+A\,\overline{B}\,\overline{C} \\
&= A\,\overline{B}(C+\overline{C}) \\
&= A\,\overline{B}
\end{aligned}
$$

[**例 1.2.5**]　化简函数 $F=AB+\overline{A}C+\overline{B}C$。

[**解**]　由分配律可得：

$$
\begin{aligned}
F &= AB+\overline{A}C+\overline{B}C \\
&= AB+(\overline{A}+\overline{B})C
\end{aligned}
$$

由反演律可得：

$$F=AB+\overline{AB}C$$

由吸收律可得：

$$F=AB+C$$

［例 1.2.6］ 化简函数 $F=AC+\overline{C}D+ADE$。

［解］ 由交换律和包含律可得：

$$\begin{aligned}F&=AC+\overline{C}D+ADE\\&=CA+\overline{C}D+ADE\\&=CA+\overline{C}D\\&=AC+\overline{C}D\end{aligned}$$

公式化简法的优点：这种方法不仅可以帮助我们更加熟悉逻辑代数的基本原理和公式，并且对于实际逻辑电路的设计也是很有用处的；在某些情况下使用起来很简便，特别是当变量较多时，这一优点体现得更加明显。例如：

$$A+ABC\overline{D}E=A$$

$$AC+\overline{A}CDE+BCDEFG=AC+\overline{A}CDE$$

公式化简法的缺点：要求能灵活运用逻辑代数的基本定律和准则；由于化简过程因人而异，因而没有明确的、规律的化简步骤，因此不便于通过计算机自动实现逻辑函数的化简；此外，公式化简法有时也不容易判断化简结果是否最简。

1.2.5 逻辑函数的卡诺图化简法

1. 逻辑函数的最小项

视频

最小项表达式

设 A、B、C 是 3 个逻辑变量，这 3 个逻辑变量可构成许多乘积项，如 $A\overline{B}\,\overline{C}$，$\overline{A}(B+C)$，$AB\overline{C}$等。其中有一类特殊的乘积项，它们是 $\overline{A}\,\overline{B}\,\overline{C}$、$\overline{A}\,\overline{B}C$、$\overline{A}B\overline{C}$、$\overline{A}BC$、$A\overline{B}\,\overline{C}$、$A\overline{B}C$、$AB\overline{C}$、$ABC$。

这 8 个乘积项的特点如下。

(1)每项都包含这 3 个变量。

(2)每个变量都以原变量 A、B、C 或以反变量 $\overline{A}$、$\overline{B}$、$\overline{C}$ 的形式出现，但同一变量的原变量和反变量不会同时出现在一项中。

这 8 个乘积项称为变量 A、B、C 的最小项。除了这 8 项，其余项如 AB、$A+B\overline{C}$、AB、$\overline{B}C$ 等都不是最小项。

如果逻辑变量的个数为 n，那么其最小项个数为 2^n。

为了书写方便，我们对最小项进行编号。每个最小项对应的编号是 m_i。以三变量 A、B、C 为例，它的 8 个最小项所对应的编号如下。

$\overline{A}\,\overline{B}\,\overline{C}=000$，$m_0=0$

$\overline{A}\,\overline{B}C=001$，$m_1=1$

$\overline{A}B\,\overline{C}=010$，$m_2=2$

$\overline{A}BC=011$，$m_3=3$

$A\overline{B}\,\overline{C}=100$，$m_4=4$

$A\overline{B}C=101$，$m_5=5$

$AB\overline{C}=110$，$m_6=6$

$ABC=111$，$m_7=7$

编号的方法是：当乘积项中的变量的次序确定后，如按 A、B、C 次序，乘积项中的原变量记为 1，反变量记为 0，例如，$AB\overline{C}$记为 110，对应的二进制编号是 110，十进制编号为 6，即$m_6=6$。

2. 逻辑函数的最小项表达式

一个逻辑函数的表达式不是唯一的。当它被表示成最小项之和时，这时的表达式就称为逻辑函数的最小项表达式。

例如，逻辑函数 $F(A,B,C)=AB+\overline{A}C$，利用逻辑代数的基本公式，我们可以将它写成如下形式：

$$F=AB+\overline{A}C=AB(C+\overline{C})+\overline{A}(B+\overline{B})C=ABC+AB\overline{C}+\overline{A}BC+\overline{A}\,\overline{B}C$$

此式由 4 个最小项组成，这个由最小项之和构成的表达式就是函数 $F(A,B,C)$的最小项表达式。这个表达式也可写成：

$$F(A,B,C)=m_7+m_6+m_3+m_1=m_1+m_3+m_6+m_7$$

为了简化书写，这个表达式可写成：

$$F(A,B,C)=\sum m\,(1,3,6,7)$$

任何逻辑函数都可以化简成最小项表达式的形式，并且任何逻辑函数最小项表达式的形式都是唯一的。

将逻辑函数化简成最小项表达式形式的方法可以采用上文所述的公式法，也可以用真值表法。如函数 $F(A,B,C)=AB+\overline{A}C$，其真值表如表 1-8 所示。

表 1-8　函数 $F(A,B,C)=AB+\overline{A}C$ 的真值表

A	B	C	$F(A,B,C)=AB+\overline{A}C$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0

续表

A	B	C	$F(A,B,C)=AB+\overline{A}C$
1	0	1	0
1	1	0	1
1	1	1	1

找出真值表中所有 F 值为 1 的行，其相应的变量组合为最小项表达式中的一项。逻辑函数 $F(A,B,C)=AB+\overline{A}C$ 有 4 行 F 的值为 1，对应的变量组合分别为 $\overline{A}\,\overline{B}C$、$\overline{A}BC$、$AB\overline{C}$、$ABC$，所以其最小项表达式为

$$F=\overline{A}\,\overline{B}C+\overline{A}BC+AB\overline{C}+ABC=m_1+m_3+m_6+m_7=\sum m(1,3,6,7)$$

3. 卡诺图

视频

卡诺图绘制

逻辑函数的卡诺图就是将逻辑函数的最小项表达式中的各个最小项相应地填入一个特定的方格图内，这个方格图就是卡诺图。因此卡诺图是逻辑函数的一种图形表示法。下面介绍卡诺图的画法。

（1）因为 n 个变量有 2^n 个最小项，所以我们首先画一个矩形，将这个矩形分成 2^n 个小格。

（2）每个小格按最小项 m_i 编号，图 1-3 画出了双变量、三变量、四变量和五变量的卡诺图及其编号。

为了说明编号的方法，我们以三变量为例。因为 $2^3=8$，所以三变量对应的方格数为 8，因为变量 AB 可能的取值有 00、01、11、10，而 C 的取值可能有 0 或 1，因此第一行第一列小方格对应着 $A=0,B=0,C=0$，即 $\overline{A}\,\overline{B}\,\overline{C}$，故其编号为 m_0，记为 0；第一行第二列小方格对应着 $A=0$，$B=1,C=0$，即 $\overline{A}B\overline{C}$，其编号为 m_2，记为 2。其他的编号依次类推，这里就不再一一说明了。

B \ A	0	1
0	0	2
1	1	3

(a) 双变量

C \ AB	00	01	11	10
0	0	2	6	4
1	1	3	7	5

(b) 三变量

CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

(c) 四变量

DE \ ABC	000	001	011	010	100	101	111	110
00	0	4	12	8	16	20	28	24
01	1	5	13	9	17	21	29	25
11	3	7	15	11	19	23	31	27
10	2	6	14	10	18	22	30	26

(d) 五变量

图 1-3　不同变量的卡诺图

在编号时还有一个原则，那就是相邻两个方格的二进制编号只能有一位不同。例如，双变量第一行第一列小方格的二进制编号为 00，第一行第二列小方格的二进制编号为 01，这两个二进制编号的第一位相同，都是 0，第二位不同。再如，三变量编号为 m_6 的小方格与编号为 m_4 的方格，前者的二进制编号是 110，后者的二进制编号是 100，只有第二位不同。

(3)小方格编号完成后，就可根据逻辑函数的最小项表达式，将表达式中存在的最小项填入相应的方格中，而不存在的最小项则略去不填。例如：

$$F(A,B,C)=m_1+m_3+m_6+m_7=\sum m(1,3,6,7)$$

$F(A,B,C)$ 相应的卡诺图如图 1-4 所示。

C \ AB	00	01	11	10
0			6	
1	1	3	7	

图 1-4　$\sum m(1,3,6,7)$ 的卡诺图

视频

卡诺图化简案例分析

4. 利用卡诺图化简逻辑函数的基本原理和方法

用卡诺图对逻辑函数进行化简的出发点是最小项表达式，化简的目标是最简表达式，通常是最简“与或”表达式，化简的工具则是逻辑函数的卡诺图。下面将介绍如何用卡诺图对逻辑函数进行简化。

1）利用卡诺图化简逻辑函数的基本原理

利用卡诺图化简逻辑函数的基本原理是公式 $AB+A\overline{B}=A$。在此式中，两个乘积项被合并成一项。相同的因子 A 被保留下来，而互补因子 B、$\overline{B}$ 则被消去了。由于卡诺图编号的原则是相邻方格的二进制编号只能有一位不同，因此我们可以依据上面的这个公式，对相邻项进行化简。例如：

$$F(A,B,C)=\overline{A}\,\overline{B}C+\overline{A}B\overline{C}+AB\overline{C}=m_1+m_2+m_6$$

其相应的卡诺图如图 1-5 所示。

C \ AB	00	01	11	10
0		2	6	
1	1			

图 1-5　$\sum m(1,2,6)$ 的卡诺图

编号为 m_2 的方格与编号为 m_6 的方格相邻，对应的最小项分别为 $\overline{A}B\,\overline{C}$ 和 $AB\,\overline{C}$，因为

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

所以这两项可以合并为一项，相同的因子 B、$\overline{C}$ 被保留下来，而不同的因子 A、$\overline{A}$ 则被消

去了。编号为 m_1 的方格无合并项，所以该项保留，化简结果为 $F(A,B,C)=\overline{A}\,\overline{B}C+B\overline{C}$。

再如，四变量逻辑函数 $F(A,B,C,D)=\overline{A}B\,\overline{C}D+AB\,\overline{C}D+\overline{A}BCD+ABCD$，其卡诺图如图 1-6 所示。

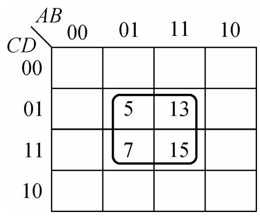


图 1-6　$\sum m(5,7,13,15)$ 的卡诺图

编号为 m_5、m_{13}、m_7、m_{15} 的四个相邻方格对应的最小项分别为 $\overline{A}B\,\overline{C}D$，$AB\,\overline{C}D$，$\overline{A}BCD$，$ABCD$。由于

$$F=\overline{A}B\overline{C}D+AB\overline{C}D+\overline{A}BCD+ABCD=(A+\overline{A})B\overline{C}D+(A+\overline{A})BCD$$

$$=B\overline{C}D+BCD=BD$$

所以化简结果 $F(A,B,C,D)=BD$。四项被合并成一项。只保留了相同的因子 B、D。

下面介绍如何利用卡诺图对逻辑函数进行化简。

2)利用卡诺图化简逻辑函数的方法

在已知逻辑函数的最小项表达式并画出逻辑函数的卡诺图后，我们就要：

(1)在卡诺图上以 1，2，4，8，…，2^n 为一组，将相邻项圈起来。

(2)对相邻项进行合并。合并的方法是保留相邻项中相同的因子，舍弃不同的因子。

(3)将合并结果相加，即得最简“与或”表达式。

在利用卡诺图化简逻辑函数时，需要注意以下几个问题。

(1)相邻项是指二进制编号只有一位不同的最小项。

当逻辑变量较多时(超过六个)，卡诺图的直观性就变差了，所以就很少使用了。

(2)圈的面积越大，消去的变量越多，即乘积项越简。两项合并可以消去一个变量，四项合并可消去两个变量，八项合并可消去三个变量。

(3)圈的数目越多，化简得到的乘积项的数目越少。

(4)每画一个圈，都至少包含一个新的最小项。

(5)一个最小项可以被重复使用，但至少要被使用一次。

(6)当所有最小项都被圈完时，化简结束。

[例 1.2.7]　用卡诺图化简逻辑函数：

$$F(A,B,C,D)=\sum m(1,2,4,6,9)$$

［解］　第一步，画出逻辑函数的卡诺图并将相邻项圈起来，如图 1-7 所示。

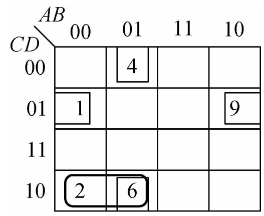


图 1-7　$\sum m(1,2,4,6,9)$的卡诺图化简

第二步，对相邻项进行合并。由图 1-7 可知，m_1号和 m_9号相邻，合并结果为 $\overline{B}\,\overline{C}D$；$m_4$号和 m_6号相邻，合并结果为 $\overline{A}\,B\overline{D}$；$m_2$号和 m_6号相邻，合并结果为 $\overline{A}\,C\overline{D}$。

第三步，将合并结果相加，获得最简“与或”表达式：

$$F(A,B,C,D)=\overline{B}\,\overline{C}D+\overline{A}C\overline{D}+\overline{A}B\overline{D}$$

这里，m_6号被使用两次，但第二次被使用时包含了一个新的最小项 m_2号或 m_4号。

［**例 1.2.8**］　用卡诺图化简逻辑函数：

$$F(A,B,C,D)=\sum m\,(0,1,3,8,9,11,13,14)$$

［解］　第一步，画出逻辑函数的卡诺图并将相邻项圈起来，如图 1-8 所示。

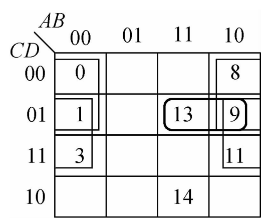


图 1-8　$\sum m(0,1,3,8,9,11,13,14)$的卡诺图化简

第二步，对相邻项进行合并。由图 1-8 可知：

m_0，m_1，m_8，m_9四项相邻，因此合并结果为 $\overline{B}\,\overline{C}$；

m_1，m_3，m_9，m_{11}四项相邻，因此合并结果为 $\overline{B}D$；

m_{13}，m_9两项相邻，因此合并结果为 $A\,\overline{C}D$；

m_{14}号没有相邻项，所以合并结果是 $ABC\,\overline{D}$。

第三步，将合并结果相加，获得最简“与或”表达式：

$$F(A,B,C,D)=\overline{B}\,\overline{C}+\overline{B}D+A\overline{C}D+ABC\overline{D}$$

［**例 1.2.9**］　用卡诺图化简逻辑函数：

$$F(A,B,C,D)=A\overline{C}\,\overline{D}+A\overline{B}CD+\overline{B}CD+A\overline{B}CD+A\overline{B}C$$

［解］　第一步，观察逻辑函数的表达式可以看出，这里给出的表达式并不是最小项表达

式，所以应先将其改写为最小项表达式。但有时为了简便，也可以直接在卡诺图中标出 F 所包含的最小项。本例中，我们采用直接标出最小项的方法。

采用直接标出最小项的方法画出逻辑函数的卡诺图，如图 1-9 所示。

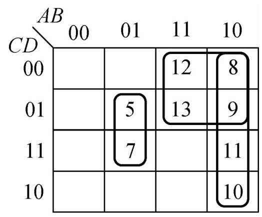


图 1-9　直接标出最小项的卡诺图

$A\overline{C}\,\overline{D}$ 含有最小项 $AB\overline{C}\,\overline{D}$ 和 $A\overline{B}\,\overline{C}\,\overline{D}$，其在卡诺图上分别为 m_{12}、m_8 号。

$A\overline{B}\,\overline{C}D$ 在卡诺图上为 m_9 号。

$B\overline{C}D$ 含有最小项 $AB\overline{C}D$ 和 $\overline{A}B\overline{C}D$，其在卡诺图上分别为 m_{13}、m_5 号。

$\overline{A}BCD$ 在卡诺图上为 m_7 号。

$A\overline{B}C$ 含有最小项 $A\overline{B}CD$ 和 $A\overline{B}C\overline{D}$，其在卡诺图上分别为 m_{11}、m_{10} 号。

第二步，对相邻项进行合并。由图 1-9 可知：

m_8，m_9，m_{11}，m_{10} 四项相邻，因此合并结果为 $A\overline{B}$。

m_{12}，m_8，m_{13}，m_9 四项相邻，因此合并结果为 $A\overline{C}$。

m_5，m_7 两项相邻，因此合并结果为 $\overline{A}BD$。

第三步，将合并结果相加，获得最简"与或"表达式：

$$F(A,B,C,D)=A\overline{B}+A\overline{C}+\overline{A}BD$$

[例 1.2.10]　用卡诺图化简逻辑函数：

$$F(A,B,C,D)=\sum m\,(0,1,2,5,8,10,15)$$

[解]　第一步，画出逻辑函数的卡诺图，并将相邻项圈起来，如图 1-10 所示。

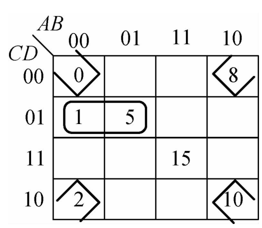


图 1-10　$\sum m\,(0,1,2,5,8,10,15)$的卡诺图化简

第二步，对相邻项进行合并。由图 1-10 可知：

m_0, m_8, m_2, m_{10} 四项相邻，因此合并结果为 $\overline{B}\,\overline{D}$。

m_1, m_5 两项相邻，因此合并结果为 $\overline{A}\,\overline{C}D$。

m_{15} 号没有相邻项，所以合并结果是 $ABCD$。

第三步，将合并结果相加，获得最简"与或"表达式：

$$F(A,B,C,D)=\overline{B}\,\overline{D}+\overline{A}\,\overline{C}D+ABCD$$

5. 含有无关最小项的逻辑函数的卡诺图化简法

在实际问题中，有时某些变量的取值会受到实际逻辑问题的限制，即某些变量的取值不可能出现，或者对结果没有影响，这些变量的取值所对应的最小项就称为无关最小项或任意项。

［**例 1.2.11**］ 有一个"四舍五入"逻辑电路，如图 1-11 所示。输入十进制数 X 按"8421"编码，即 $X=8A+4B+2C+D\leqslant 9$。要求当 $X\geqslant 5$ 时，输出 $Y=1$；否则，输出 $Y=0$。

图 1-11 例 1.2.11 的"四舍五入"逻辑电路图

［**解**］ 根据题意，列出真值表，如表 1-9 所示。

表 1-9 "四舍五入"逻辑电路真值表

X	A	B	C	D	Y
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
—	1	0	1	0	d
—	1	0	1	1	d
—	1	1	0	0	d
—	1	1	0	1	d
—	1	1	1	0	d
—	1	1	1	1	d

在本电路中，由于输入变量 A、B、C、D 的后六种组合（1010～1111）是不可能出现的，因此它们所对应的 Y 值也是没有意义的，即这 6 个最小项是无关最小项，它们对应的 Y 值是 1 或 0 都无关紧要。

为了便于化简逻辑函数，我们在真值表中仍然列出这 6 种组合，并把它们对应的 Y 值记为 d。我们可以这样来理解 d：表示 Y 的值既可以是 1，也可以是 0。

这样，Y 的表达式可以写为

$$Y = \sum m(5,6,7,8,9) + \sum d(10,11,12,13,14,15)$$

如果不利用无关最小项，那么根据卡诺图化简法，如图 1-12 所示，可得：

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

CD \ AB	00	01	11	10
00				8
01		5		9
11		7		
10		6		

图 1-12　不利用无关最小项的卡诺图化简

但是如果把无关最小项考虑进去，情况就不同了。如图 1-13 所示，在无关最小项的方格内填写 d，我们既可以将它们看作 1，也可以将它们看作 0，在这里我们将它们看作 1，根据卡诺图化简法，可得：

$$Y = A + BC + BD$$

CD \ AB	00	01	11	10
00			d	8
01		5	d	9
11		7	d	d
10		6	d	d

图 1-13　利用无关最小项的卡诺图化简

可见，考虑无关最小项与不考虑无关最小项的化简结果是很不一样的。这说明通过恰当选择无关最小项，往往可以得到较简单的逻辑函数表达式。

课后练习

1.1　进位制中的基数与权值分别指什么？

1.2　简述二进制、八进制与十六进制的特点及相互转换方法。

1.3　将下列二进制数转换成十进制数、八进制数和十六进制数。

$(1100)_2=(\quad)_{10}=(\quad)_8=(\quad)_{16}$

$(10101101)_2=(\quad)_{10}=(\quad)_8=(\quad)_{16}$

$(11111111)_2=(\quad)_{10}=(\quad)_8=(\quad)_{16}$

$(1010.0101)_2=(\quad)_{10}=(\quad)_8=(\quad)_{16}$

1.4　将下列十进制数转换成二进制数、八进制数和十六进制数。

$(98)_{10}=(\quad)_2=(\quad)_8=(\quad)_{16}$

$(64)_{10}=(\quad)_2=(\quad)_8=(\quad)_{16}$

$(128)_{10}=(\quad)_2=(\quad)_8=(\quad)_{16}$

$(4095)_{10}=(\quad)_2=(\quad)_8=(\quad)_{16}$

$(32.3125)_{10}=(\quad)_2=(\quad)_8=(\quad)_{16}$

1.5　对如下数据进行转换。

(1)$(371.236)_8=(\quad)_{16}=(\quad)_2$

(2)$(48A)_{16}=(\quad)_8=(\quad)_{10}$

1.6　逻辑代数与普通代数有哪些主要区别?

1.7　什么叫真值表?试写出利用两个变量进行“与”运算,“或”运算及“非”运算的真值表。

1.8　列出下列两组逻辑函数的真值表,并说明各组 F_1 和 F_2 有何关系。

$$(1)\begin{cases}F_1=ABC+\overline{A}\,\overline{B}\,\overline{C}\\ F_2=\overline{A\overline{B}+B\overline{C}+C\overline{A}}\end{cases}$$

$$(2)\begin{cases}F_1=A\overline{B}+B\overline{C}+C\overline{A}\\ F_2=\overline{A}B+\overline{B}C+\overline{C}A\end{cases}$$

1.9　写出下列表达式的对偶式。

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

(2)$F=\overline{A\,\overline{BC}}(\overline{A}+C)$

1.10　利用反演规则求下列逻辑函数的反函数。

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

(2)$F=A\,\overline{B}+\overline{B}C+C(\overline{A}+B+C)$

1.11　什么是逻辑函数的最简表达式?逻辑函数的最简表达式是唯一的吗?什么是逻辑函数的最简“与或”表达式?

1.12　用公式化简法将下列逻辑函数化简为最简的“与或”表达式。

(1)$F=\overline{A}\,\overline{B}\,\overline{C}+\overline{A}BC+ABC+AB\overline{C}$

(2) $F=A\overline{B}+B+BCD$

(3) $F=\overline{A}B+\overline{A}C+\overline{B}\,\overline{C}+AD$

(4) $F=\overline{A}\,\overline{B}+\overline{A}\,\overline{C}D+AC+B\overline{C}$

(5) $F=A(B+\overline{C})+\overline{A}(\overline{B}+C)+BCD+\overline{B}\,\overline{C}D$

1.13　什么叫最小项？最小项有哪些性质？简述最小项的编号方法。

1.14　什么是逻辑函数的最小项表达式？逻辑函数的最小项表达式是唯一的吗？它与一般的“与或”表达式有什么区别？如何用真值表法或公式化简法将逻辑函数化简为最小项表达式形式？

1.15　什么叫卡诺图？卡诺图方格的编号原则是什么？

1.16　用卡诺图化简法化简逻辑函数主要分为哪几步？主要应注意哪几点？

1.17　迄今为止，你已经掌握了几种逻辑函数的化简方法？各自的主要特点是什么？

1.18　什么是无关最小项？怎样进行含有无关最小项逻辑函数的化简？

1.19　用卡诺图化简法化简下列逻辑函数。

(1) $F(A,B,C,D)=\sum m(0,1,4,6,9,13,14,15)$

(2) $F(A,B,C,D,E)=\sum m(3,4,6,9,11,13,15,18,25,26,27,29,31)$

(3) $F(A,B,C,D)=\sum m(2,9,10,12,13)+\sum d(1,5,14)$

(4) $F(A,B,C,D)=\sum m(0,1,3,5,8,9)$

(5) $F(A,B,C,D)=B\overline{C}D+A\overline{B}\,\overline{C}D$，其中 $C+D=0$ 不可能出现。

1.20　下列逻辑函数是最简“与或”表达式吗？如果不是，请化简。

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

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

(3) $F=ABD+\overline{A}B\overline{D}+A\overline{C}\,\overline{D}+\overline{A}\,\overline{C}D+B\overline{C}$

(4) $F=A\overline{B}+C\overline{D}+\overline{A}BC+ABD$