普通高等院校计算机专业(本科)实用教程系列

数字逻辑实用教程

王玉龙　编著

清 华 大 学 出 版 社

(京) 新登字 158 号

内 容 简 介

本书是高等院校计算机专业实用系列教材之一，系统介绍了数字电路逻辑设计的基本理论和方法。全书共 8 章，第 1 章介绍了数字电路逻辑设计的数学工具——逻辑代数，第 2 ～5 章介绍了组合逻辑线路和时序逻辑线路的分析与设计方法，第 6 章介绍了可编程逻辑器件用于数字设计的基本原理和方法，第 7 章介绍了数字电路逻辑设计的基本实验方法，第 8 章提供了本书前六章练习题的解答。

本书是作者在长期从事数字逻辑教学工作的基础上编写的，具有易读、简明和实用等特点。本书取材较新、概念清晰、逻辑性强、语言流畅，适于读者自学。

本书可作为高等院校计算机专业本科“ 数字逻辑 ”课程的教材。若删去书中带 * 号的内容，本书也可作为计算机专业大专生的“ 数字逻辑 ”(或数字电路) 课程的教材。本书也可供从事计算机、自动化及电子学等专业的科技人员参考。

书　　名: 数字逻辑实用教程
作　　者: 王玉龙
出 版 者: 清华大学出版社(北京清华大学学研大厦，邮编: 100084)
　　http: //www.tup.tsinghua.edu.cn
责任编辑: 闫红梅
印 刷 者: 清华大学印刷厂
发 行 者: 新华书店总店北京发行所
开　　本: 787× 1092 1/16　印张: 19.75　字数: 480 千字
版　　次: 2002 年 3 月第 1 版　2002 年 3 月第 1 次印刷
书　　号: ISBN 7-302-05127-5/TP・3004
印　　数: 0001 ～5000
定　　价: 28.00 元

序　　言

时光更迭、历史嬗递。中国经济带着她足以令世人惊叹的持续高速发展驶入了一个新的世纪，一个新的千年。世纪之初，以微电子、计算机、软件、通信技术为主导的信息技术革命给我们生存的社会所带来的变化令人目不暇接。软件是优化我国产业结构、加速传统产业改造和用信息化带动工业化的基础产业，是体现国家竞争力的战略性产业，是从事知识的提炼、总结、深化和应用的高智型产业；软件关系到国家的安全，是保证我国政治独立、文化不受侵蚀的重要因素；软件也是促进其他学科发展和提升的基础学科；软件作为 20 世纪人类文明进步的最伟大成果之一，代表了先进文化的前进方向。美国政府早在 1992 年“国家关键技术”一文中提出“美国在软件开发和应用上所处的传统领先地位是信息技术及其他重要领域竞争能力的一个关键因素”，“一个成熟的软件制造工业的发展是满足商业与国防对复杂程序日益增长的要求所必需的”，“在很多国家关键技术中，软件是关键的起推动作用(或阻碍作用)的因素”。在 1999 年 1 月美国总统信息技术顾问委员会的报告“21 世纪的信息技术”中指出“从台式计算机、电话系统到股市，我们的经济与社会越来越依赖于软件”，“软件研究为基础研究方面最优先发展的领域”。而软件人才的缺乏和激烈竞争是当前国际的共性问题。各国、各企业都对培养、引进软件人才采取了特殊政策与措施。

为了满足社会对软件人才的需要，为了让更多的人可以更快地学到实用的软件理论、技术与方法，我们编著了《普通高等院校计算机专业(本科)实用教程系列丛书》。本套丛书面向普通高等院校学生，以培养面向 21 世纪计算机专业应用人才(以软件工程师为主)为目标，以简明实用、便于学习、反映计算机技术最新发展和应用为特色，具体归纳为以下几点：

1．讲透基本理论、基本原理、方法和技术，在写法上力求叙述详细，算法具体，通俗易懂，便于自学。

2．理论结合实际。计算机是一门实践性很强的科学，丛书贯彻从实践中来到实践中去的原则，许多技术理论结合实例讲，以便于学习和理解。

3．本丛书形成完整的体系，每本教材既有相对独立性，又有相互衔接和呼应，为总的培养目标服务。

4．每本教材都配以习题和实验，在各教学阶段安排课程设计或大作业，培养学生的实战能力与创新精神。习题和实验可以制作成光盘。

新世纪曙光激人向上，催人奋进。江总书记在十五届五中全会上的讲话：“大力推进国民经济和社会信息化，是覆盖现代化建设全局的战略举措。以信息化带动工业化，发挥优势，实现社会生产力的跨越式发展。”指明了我国信息界前进的方向。21 世纪日趋开放的国策与更加迅速发展的科技会托起祖国更加辉煌灿烂的明天。

孙家广

2001 年 3 月

普通高等院校计算机专业(本科)实用教程
系列丛书编委会

前　言

数字逻辑”是数字电路逻辑设计的简称，其内容是讲述应用数字电路进行数字系统逻辑设计(简称数字设计)的方法。随着微电子技术及计算机硬件技术的迅速发展，数字设计所使用的器件发生了很大变化：从最初的分立元件、中小规模集成电路到大规模、超大规模集成电路，从标准的通用芯片到可编程逻辑器件(PLD)。这一变化导致数字设计的方法不断更新，使数字系统的逻辑设计从“纯硬件”设计演变为借助于软件工具来完成硬件设计。尽管如此，数字电路逻辑设计的基础理论与基本原理仍没有改变。数字逻辑领域这一“变”与“不变”的发展趋势，要求本课程在讲授数字逻辑的基础理论与基本原理的同时，培养学生具有挑战数字逻辑新技术的能力。另一方面，数字系统中超大规模集成电路的广泛应用，使计算机专业的大多数学生已无须直接参与逻辑部件(或数字系统)的设计、制造与调试，而只是“拿来就用”。这一现实使本课程的主要任务已明显地转化为为学习后续课程“计算机组成原理”等打下基础。此外，本课程的知识结构特点有助于训练学生的逻辑思维能力、运用形式化方法描述客观世界的能力以及使用计算机硬件的实践能力。本书就是在分析数字逻辑这门课程上述背景材料的基础上而编写的一本实用教程。

本书包括下列八章内容，第1章：逻辑代数基础；第2章：组合线路的分析；第3章：组合线路的设计；第4章：时序线路的分析；第5章：时序线路的设计；第6章：可编程逻辑器件；第7章：数字逻辑实验指南；第8章：练习题解，主要是各章练习题的部分题解。本书第1～6章是数字逻辑的主教材，内容简明扼要，符合教学大纲要求；第7、8章是数字逻辑的辅助教材，介绍了该课程的六个基本实验的内容，给出了全书各章练习题的大部分题解，以便于学生自行设计实验方案和检查做题的正确性。

本书着眼于培养学生分析问题和解决问题的能力。在内容组织上，以讲述“方法”为重点，力求为学生提供独立分析和设计逻辑线路的“工具”，而不是向学生灌输各种各样的逻辑线路；在讲述方法上，力求对每一个求解的问题作出思路分析，尽量避免就事论事，以使学生了解其来龙去脉；在文字叙述上，力求做到说理清楚、深入浅出，便于学生自学。

本书是作者在长期从事数字电路逻辑设计的教学工作基础上编写的，通过对上述诸方面的努力，使本书具有较好的实用性、简明性和易读性。这些特点使本书很适用于高校计算机专业本科的“数字逻辑”(或数字电路)课程的教材。书中带*号的内容可根据教学要求及教学学时数删去某些章节，或指定为选学内容。

在本书的编写过程中，曾得到本系列教程编委会老师的指导和帮助；也得到北方工业大学吴乐明老师的大力帮助，她为全书进行了录入、整理和校对，付出了辛勤的劳动。在此，对他(她)们表示衷心的感谢。对本书中尚可能出现的错误和不妥之处，敬请读者批评指正。

王玉龙

2001年9月

目　　录

第 1 章　逻辑代数基础

众所周知，电子计算机归根到底是对“0”和“1”进行处理，它们是通过电子开关线路(如门电路、触发器等)实现的。这些开关电路具有下列基本特点：从线路内部看，或是管子导通，或是管子截止；从线路的输入输出看，或是高电平，或是低电平。这种开关电路的工作状态可以用二元布尔代数描述，通常又称为开关代数，或逻辑代数。

本章将从实用的角度介绍逻辑代数的基础知识，以使读者掌握分析和设计数字逻辑网络所需要的数学工具。下面将先介绍逻辑代数的变量及其基本运算，然后介绍逻辑代数的函数及其表示法，以及逻辑代数中的常用公式和定理，最后介绍逻辑函数的化简方法。

1.1　逻辑变量及其基本运算

逻辑代数是一个由逻辑变量集 K，常量 0、1 及“或”、“与”、“非”三种运算符所构成的代数系统，记为(K，+，·，-，0，1)。其中逻辑变量集是指逻辑代数中的所有可能变量的集合，它可用任何字母表示，但每一个变量的取值只可能为常量 0 或 1。而且，逻辑代数中的变量只有三种运算，即“或”运算、“与”运算及“非”运算。其定义如下：

“或”运算

+	0	1
0	0	1
1	1	1

“与”运算

·	0	1
0	0	0
1	0	1

“非”运算

-	
0	1
1	0

显而易见，逻辑代数是一种比普通代数简单得多的代数系统。例如，普通代数中的变量取值可为负无穷大到正无穷大之间的任意数，而逻辑代数中的变量取值只能为 0 或 1；普通代数中的变量运算包括加、减、乘、除、乘方、开方等许多种，而逻辑代数中的变量运算只有“或”、“与”、“非”三种。但是，这种简单的逻辑代数却能描述数字系统中任何复杂的逻辑网络。这是因为不管逻辑网络多么复杂，总是可认为由“或”、“与”、“非”等简单门电路组成，而这些门电路的输入输出信号可看作为逻辑变量，输出与输入信号之间的关系可用“或”、“与”、“非”三种运算描述。这里，我们也不难理解，逻辑代数中的“0”、“1”与普通代数中的 0、1 含义是不同的。逻辑代数的 0、1 表示了信号的“无”、“有”，或命题的“假”、“真”。

根据逻辑变量的取值只有 0 和 1，及逻辑变量仅有的三种运算的定义，不难推出下列基本公式，或称为逻辑代数的公理。

0 - 1 律:

$$A + 0 = A \tag{1.1}$$

$$A + 1 = 1$$

$$A \cdot 1 = A$$

$$A \cdot 0 = 0 \tag{1.2}$$

重叠律:

$$A + A = A \tag{1.3}$$

$$A \cdot A = A \tag{1.4}$$

互补律:

$$A + \bar{A} = 1 \tag{1.5}$$

$$A \cdot \bar{A} = 0 \tag{1.6}$$

对合律:

$$\bar{\bar{A}} = A \tag{1.7}$$

交换律:

$$A + B = B + A \tag{1.8}$$

$$A \cdot B = B \cdot A \tag{1.9}$$

结合律:

$$(A + B) + C = A + (B + C) \tag{1.10}$$

$$(A \cdot B) \cdot C = A \cdot (B \cdot C) \tag{1.11}$$

分配律:

$$A \cdot (B + C) = A \cdot B + A \cdot C \tag{1.12}$$

$$A + BC = (A + B)(A + C) \tag{1.13}$$

上述七组基本公式中，A、B 和 C 均为逻辑变量，且每组公式中的两个公式互为“对偶”。即将其中一式中的“+”换成“·”，“·”换成“+”，0 换成 1，1 换成 0，便得到与其相应的另一公式。

上述公式的证明是显然的，这里仅对式(1.3)及式(1.13)作一证明。

(1) 式(1.3)的证明：A 只有 0、1 两种取值，故

当 A = 1 时，$A + A = 1 + 1 = 1 = A$

当 A = 0 时，$A + A = 0 + 0 = 0 = A$

即证得不论 A 为 0 或 1，均有 $A + A = A$。

(2) 式(1.13)的证明：证明的前提是假定其前的基本公式已获证，故有

$$\begin{aligned} A + BC &= A(1 + B + C) + BC && \text{0 - 1 律} \\ &= A + AB + AC + BC && \text{分配律} \\ &= AA + AB + AC + BC && \text{重叠律} \\ &= (AA + AC) + (AB + BC) && \text{交换律、结合律} \\ &= A(A + C) + B(A + C) && \text{分配律} \\ &= (A + B)(A + C) \end{aligned}$$

不难看出，上述基本公式中，某些公式与普通代数中的公式相同，但某些公式却是逻

辑代数中所特有的。这里需着重指出的是式(1.13)，称为加对乘的分配，这一公式在普通代数中是不成立的，它是逻辑代数中特有的非常有用的公式。

1.2 逻辑函数及其标准形式

这一节将先给出逻辑函数的定义，然后讨论逻辑函数的三种表示形式，最后介绍逻辑函数的两种标准形式(最小项表达式及最大项表达式)及其性质。

1.2.1 逻辑函数的定义

逻辑代数中的函数定义与普通代数中的函数定义极为相似，可叙述如下:

设某一逻辑网络的输入逻辑变量为 A_1，A_2，…，A_n，输出逻辑变量为 F，如图 1.1 所示。当 A_1，A_2，…，A_n 的取值确定后，F 的值就惟一地被确定下来，则称 F 是 A_1，A_2，…，A_n 的逻辑函数，记为

$$F = f(A_1, A_2, \cdots, A_n) \tag{1.14}$$

逻辑变量与逻辑函数的取值都只可能是 0 或 1，但相对某一逻辑网络而言，逻辑变量的取值是“自行”变化的，而逻辑函数的取值则是由逻辑变量的取值和网络本身的结构决定的。

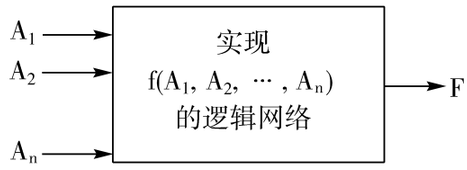


图 1.1 $F = f(A_1, A_2, \cdots, A_n)$

1.2.2 逻辑函数的表示法

表示逻辑函数的方法有三种：逻辑表达式、真值表和卡诺图。这与普通代数中用公式、表格和图形方法表示一个函数相类似，下面分别说明这三种方法。

1. 逻辑表达式

逻辑表达式是由逻辑变量和“或”、“与”、“非”三种运算符所构成的式子，这是一种用公式表示逻辑函数的方法。若要表示这样的一个函数关系：当两个逻辑变量(A 和 B)取值相同时，逻辑函数的取值为“1”；否则，逻辑函数的取值为“0”。可以用下列逻辑表达式:

$$F = f(A, B) = AB + \overline{A}\,\overline{B} \tag{1.15}$$

我们暂且不讨论该式是怎样得到的，但可验证它是正确的。因为当 A 和 B 取值相同时，A 和 B 都同时为 1 或同时为 0，代入式(1.15) 必得 F = 1。反之，当 A 和 B 取值不同时，A 和 B 中必有一个为“1”，另一个为“0”，代入式(1.15) 必得 F = 0。

2. 真值表

真值表是由逻辑变量的所有可能取值组合及其对应的逻辑函数值所构成的表格，这是一种用表格表示逻辑函数的方法。对于式(1.15) 所描述的逻辑函数，可以用表 1.1 所示的真值表表示。

表 1.1　式(1.15) 函数的真值表

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

表中列出了两个逻辑变量(A 和 B) 的所有可能的取值组合(00，01，10，11)，并列出了与它们相对应的逻辑函数(F) 之值。由表明显可知，当 A = B 时，F = 1；当 A ≠ B 时，F = 0。

上述真值表中的变量为两个，共有 2^2 种组合，故该表由 4 行组成。不难推得，当逻辑函数的变量为三个时，真值表就由 2^3 行组成；当逻辑函数的变量为 n 个时，真值表就由 2^n 行组成。显而易见，随着变量数目的增多，真值表的行数将急剧增加，表的规模就变得很庞大。

3. 卡诺图

卡诺图是由表示逻辑变量的所有可能组合的小方格所构成的图形，如图 1.2 所示。图中分别表示了单变量、双变量及三变量的卡诺图。

图 1.2(a) 为单变量 A 的卡诺图，它由两个小方格组成，上面一个表示 $\bar{A}$，下面一个表示 A。图 1.2(b) 为双变量 A、B 的卡诺图，它由横向两行和竖向两列组成，横向两行分别为 $\bar{A}$ 和 A，竖向两列分别为 $\bar{B}$ 和 B，从而得到如图所示的四个小方格，它们分别表示 $\bar{A}\bar{B}$、$A\bar{B}$、$\bar{A}B$、AB 四种变量组合。图 1.2(c) 为三变量 A、B、C 的卡诺图，它由八个小方格组成，分别表示三个变量可能有的八种组合：$\bar{A}\bar{B}\bar{C}$、…、ABC。

卡诺图是一种用图形表示逻辑函数的方法，只要在使函数值为 1 的变量组合所对应的小方格上标记 1，便得到该逻辑函数的卡诺图。例如，式(1.15) 所示函数的卡诺图如图 1.3 所示。

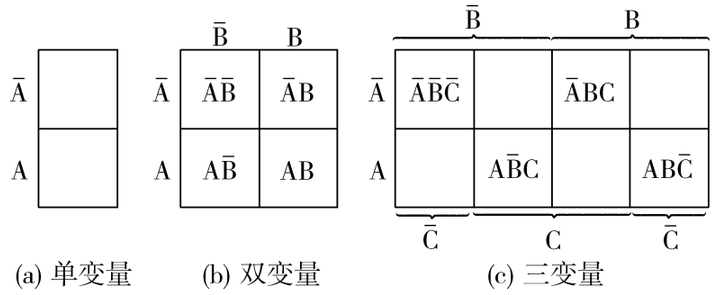


图 1.2　一至三变量的卡诺图

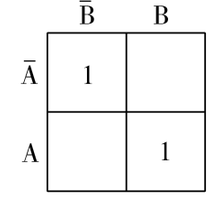


图 1.3　式(1.15) 函数的卡诺图

由上不难推得，当逻辑变量为 4 时，其卡诺图将由 2^4 个小方格组成；当逻辑变量为 n

时，其卡诺图将由 2^n 个小方格组成。

上面介绍的三种表示逻辑函数的方法，虽然各有特点，适用于不同场合，但所描述的对象却是相同的。它们之间存在内在的联系，可方便地相互变换。为了说明这一关系，先介绍一下逻辑函数的两种标准形式。

1.2.3 逻辑函数的标准形式

1. 最小项及最小项表达式

什么是最小项？为了说明这个问题，我们先来看一个简单例子。设有一个两变量的逻辑函数

$$\begin{aligned} F &= f(A, B) \\ &= A + \overline{AB} \end{aligned} \tag{1.16}$$

利用前述的基本公式，可得F 的下列形式：

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

这一事实表明，同一逻辑函数可用多种形式表示，有的比较简单，有的比较复杂。然而，在这些表达式中有一个最规则的形式，这就是上面的最后一个表达式：

$$F = \overline{A}\,\overline{B} + A\overline{B} + AB \tag{1.17}$$

该式是由若干个乘积项之和所组成，其中每个乘积项具有这样的特点：它包含有该函数的全部逻辑变量(两个)，或以原变量(A, B) 出现，或以反变量($\overline{A}$, $\overline{B}$) 出现，且每个变量在一个乘积项中只出现一次。具有这样特点的乘积项称为最小项。一般地说，最小项的定义可叙述如下：

设有 n 个逻辑变量，它们组成的乘积项(“与”项) 中，每个变量或以原变量或以反变量形式出现一次，且仅出现一次，这个乘积项称为 n 个变量的最小项。

例如，对于两个变量(A，B) 而言，最多能构成 4 个最小项：

$$\overline{A}\,\overline{B},\quad \overline{A}B,\quad A\overline{B},\quad AB$$

对于三个变量(A，B，C) 而言，最多能构成 2^3 个最小项：

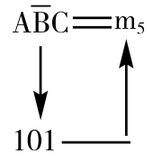
$$\overline{A}\,\overline{B}\,\overline{C},\quad \overline{A}\,\overline{B}C,\quad \overline{A}B\overline{C},\quad \overline{A}BC$$
$$A\overline{B}\,\overline{C},\quad A\overline{B}C,\quad AB\overline{C},\quad ABC$$

显然，对于 n 个变量，则可构成 2^n 个最小项。

通常，为了书写方便，最小项可用符号 m_i 表示，如三变量的 8 个最小项可表示为：

$$\overline{A}\,\overline{B}\,\overline{C} = m_0 \qquad \overline{A}\,\overline{B}C = m_1$$
$$\overline{A}B\overline{C} = m_2 \qquad \overline{A}BC = m_3$$
$$A\overline{B}\,\overline{C} = m_4 \qquad A\overline{B}C = m_5$$
$$AB\overline{C} = m_6 \qquad ABC = m_7$$

其中 m_i 的下标 i 是这样确定的：当各最小项的变量按一定次序排好后，用 1 代替其中的原变量，用 0 代替其中的反变量，便得一个二进制数，该二进制数的等值十进制数即为相应最小项符号 m_i 的 i 值。例如，上例中的 $A\bar{B}C = m_5$ 是这样得到的：



从最小项的定义出发，我们不难得知最小项的下列三个主要性质：

(1) 对于任意一个最小项，只有一组变量取值可使其值为 1。

(2) 任意两个最小项 m_i 和 m_j ($i \neq j$) 之积必为 0。

(3) n 变量的所有 2^n 个最小项之和必为 1，即

$$\sum_{i=0}^{2^n-1} m_i = 1 \qquad (1.18)$$

现在，我们进一步讨论最小项表达式的定义及其性质。

所谓最小项表达式，就是由给定函数的最小项之和所组成的逻辑表达式。如式 (1.17) 所示函数，它由三个最小项组成：

$$\begin{aligned} F &= \bar{A}\bar{B}C + A\bar{B}C + AB \\ &= m_0 + m_2 + m_3 \\ &= \sum(0, 2, 3) \end{aligned}$$

这里，借用普通代数中的“ $\sum$ ”符号表示多个最小项的累计“ 或 ”运算，圆括号内的十进制数字表示参与“ 或 ”运算的各个最小项的项号，它们就是各 m_i 的下标值。

可以证明，任何 n 变量的逻辑函数都有一个且仅有一个最小项表达式。若已知某一逻辑函数不是最小项表达式形式，则可通过反复使用下式而获得最小项表达式：

$$x = x(y + \bar{y})$$

例如，设 $F(A, B, C) = A\bar{B} + B\bar{C} + ABC$，则得

$$\begin{aligned} F(A, B, C) &= A\bar{B}(C + \bar{C}) + B\bar{C}(A + \bar{A}) + ABC \\ &= A\bar{B}C + A\bar{B}\bar{C} + AB\bar{C} + \bar{A}B\bar{C} + ABC \\ &= \bar{A}B\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + AB\bar{C} + ABC \\ &= m_2 + m_4 + m_6 + m_7 \\ &= \sum(2, 4, 6, 7) \end{aligned}$$

由该例可以看出，只要给定的函数是一积之和表达式 (也称“ 与 - 或 ”表达式)，通过对该式中的所有非最小项的乘积项(“ 与 ”项) 乘上其所缺变量之“ 原 ”加“ 反 ”，便可得到给定函数的最小项表达式。由于最小项表达式是逻辑函数的标准形式之一，故常称它为积之和范式，或主析取范式。

下面给出最小项表达式的三个主要性质，它们是

(1) 若 m_i 是逻辑函数 $F(A_1, A_2, \cdots, A_n)$ 的一个最小项，则使 $m_i = 1$ 的一组变量取值 $(a_1, a_2, \cdots, a_n)$ 必定使 F 值为 1。

(2) 若 F_1 和 F_2 都是 $A_1, A_2, \cdots, A_n$ 的函数，则 $F = F_1 + F_2$ 将包括 F_1 和 F_2 中的所

有最小项，$G = F_1 \cdot F_2$ 将包括 F_1 和 F_2 中的公有最小项。

(3) 若 $\bar{F}$ 是 F 的反函数，则 $\bar{F}$ 必定由 F 所包含的最小项之外的全部最小项所组成。

这些性质的证明较为麻烦，但通过具体例子可方便地验证这些性质的正确性。

2．最大项及最大项表达式

仿照前面给出的最小项定义，我们引入最大项的定义如下:

设有 n 个逻辑变量，它们所组成的和项(“或”项) 中，每个变量或以原变量或以反变量形式出现，且仅出现一次，这个和项称为 n 变量的最大项。

例如，对于两个变量(A，B) 而言，最多可构成 4 个最大项:

$$(\bar{A}+\bar{B}),\ (\bar{A}+B),\ (A+\bar{B}),\ (A+B)$$

对于三个变量(A，B，C) 而言，最多可构成 2^3 个最大项:

$$(\bar{A}+\bar{B}+\bar{C}),\ (\bar{A}+\bar{B}+C),\ (\bar{A}+B+\bar{C}),\ (\bar{A}+B+C)$$
$$(A+\bar{B}+\bar{C}),\ (A+\bar{B}+C),\ (A+B+\bar{C}),\ (A+B+C)$$

显然，对于 n 个变量，则可构成 2^n 个最大项。

与最小项类似，最大项可用符号 M_i 表示，但下标 i 的取值规则与最小项 i 的取值规则恰好相反。例如，最大项($A+\bar{B}+C$) 的缩写符号为 M_2，而不是 M_5，其原因如下:

$$A+\bar{B}+C = M_2$$
$$0\quad 1\quad 0$$

可见，M_i 中的下标 i 是这样确定的: 当各最大项的变量按一定次序排列好后，用 0 代替其中的原变量，用 1 代替其中的反变量，所得的二进制数的等值十进制数，便是相应最大项的符号 M_i 中的 i 值。按此规则，可以写出三变量的 8 个最大项的符号如下:

$$\bar{A}+\bar{B}+\bar{C}=M_7 \qquad \bar{A}+\bar{B}+C=M_6$$
$$\bar{A}+B+\bar{C}=M_5 \qquad \bar{A}+B+C=M_4$$
$$A+\bar{B}+\bar{C}=M_3 \qquad A+\bar{B}+C=M_2$$
$$A+B+\bar{C}=M_1 \qquad A+B+C=M_0$$

同样，最大项也具有下列三个主要性质:

(1) 对于任意一个最大项，只有一组变量取值可使其值为 0。

(2) 任意两个最大项 M_i 和 $M_j(i \neq j)$ 之和必为 1。

(3) n 变量的所有 2^n 个最大项之积必为 0，即

$$\prod_{i=0}^{2^n-1} M_i = 0 \qquad (1.19)$$

下面，我们进一步讨论最大项表达式的定义及性质。

所谓最大项表达式，就是由给定函数的最大项之积所组成的逻辑表达式。例如，函数 $F = A\bar{B}+\bar{B}\bar{C}$ 的最大项表达式为

$$\begin{aligned} F &= (\bar{A}+\bar{B}+\bar{C})(\bar{A}+B+\bar{C})(A+\bar{B}+\bar{C})(A+B+\bar{C})(A+B+C) \\ &= M_7 M_5 M_3 M_1 M_0 \\ &= \prod(0,1,3,5,7) \end{aligned}$$

这里，借用普通代数中的“$\prod$”符号表示多个最大项的累计“与”运算，圆括号中的十

进制数字是参与“ 与 ”运算的最大项的项号。

同样可以证明，任何 n 变量的逻辑函数都可展开为最大项表达式，而且这种展开是惟一的。那么，怎样把逻辑函数展开成最大项表达式呢？若已知函数为积之和形式，则需利用加对乘的分配律

$$x + yz = (x + y)(x + z)$$

将积之和表达式转换为和之积表达式(即“ 或 - 与 ”表达式)。然后，在该式的各非最大项的和项中加上它所缺变量的“ 原 ”、“ 反 ”之积(如$\times\ \bar{\times}$ 形式)，并再次使用加对乘的分配律，直到把全部和项都变为最大项，便得已知函数的最大项表达式。下面，举例说明这一方法。

例 1　已知函数

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

求取 F 的最大项表达式的过程如下:

$$\begin{aligned}
F &= (A + C)(A + B)(A + \bar{B}\bar{C}) \\
&= (A + C + B\bar{B})(A + B + C\bar{C})(A + \bar{B}\bar{C}) \\
&= (A + C + B)(A + C + \bar{B})(A + B + C)(A + B + \bar{C})(A + \bar{B}\bar{C}) \\
&= (A + B + C)(A + B + \bar{C})(A + \bar{B}\bar{C} + C)(A + \bar{B}\bar{C} + \bar{C}) \\
&= M_0 M_1 M_2 M_3 \\
&= \prod(0,1,2,3)
\end{aligned}$$

例 2　已知函数

$$F = A + \bar{B}\bar{C}$$

由下式可求得 F 的最大项表达式:

$$\begin{aligned}
F &= A + \bar{B}\bar{C} \\
&= (A + \bar{B}\bar{C})(A + BC) \\
&= 1 \cdot (A + B)(A + C) \\
&= (A + B + C\bar{C})(A + C + B\bar{B}) \\
&= (A + B + C)(A + B + \bar{C})(A + C + B)(A + C + \bar{B}) \\
&= (A + B + C)(A + B + \bar{C})(A + \bar{B}\bar{C} + C) \\
&= \prod(0,\ 1,\ 2)
\end{aligned}$$

最大项表达式是逻辑函数的另一种标准形式，通常也称为和之积范式，或主合取范式。

根据最大项表达式的定义，可推得类似于最小项表达式的三个性质，这里不再详述。

1.2.4　逻辑函数三种表示法的关系

前已指出，逻辑表达式、真值表和卡诺图是表示逻辑函数的三种手段，因而它们之间必然存在内在的联系，这一联系是通过最小项表达式实现的。