SHUZI

DIANLU YU XITONG KECHENG JINGCUI YU TIJIE

# 数字电路与系统

## 课程精粹与题解

赵　刚　等编著

四川大学出版社

# 数字电路与系统

# 课程精粹与题解

SHUZI DIANLU YU
XITONG KECHENG JINGCUI YU TIJIE

赵 刚 等编著

四川大学出版社

责任编辑:王　平
责任校对:张战清
封面设计:翼虎书装
责任印制:李　平

**图书在版编目(CIP)数据**

数字电路与系统课程精粹与题解 / 赵刚等编著. —成都:四川大学出版社, 2008.3
ISBN 978-7-5614-3973-9

Ⅰ. 数…　Ⅱ. 赵…　Ⅲ. 数字集成电路-系统设计-高等学校-教学参考资料　Ⅳ. TN431.2

中国版本图书馆 CIP 数据核字 (2008) 第 034198 号

书名　**数字电路与系统课程精粹与题解**

| | |
|---|---|
| 编　著 | 赵　刚等 |
| 出　版 | 四川大学出版社 |
| 地　址 | 成都市一环路南一段 24 号 (610065) |
| 发　行 | 四川大学出版社 |
| 书　号 | ISBN 978-7-5614-3973-9/TN·29 |
| 印　刷 | 郫县犀浦印刷厂 |
| 成品尺寸 | 185 mm×260 mm |
| 印　张 | 21 |
| 字　数 | 518 千字 |
| 版　次 | 2008 年 3 月第 1 版 |
| 印　次 | 2008 年 3 月第 1 次印刷 |
| 印　数 | 0 001～2 000 册 |
| 定　价 | 35.00 元 |

◆读者邮购本书,请与本社发行科联系。电 话:85408408/85401670/85408023　邮政编码:610065
◆本社图书如有印装质量问题,请寄回出版社调换。
◆网址:www.scupress.com.cn

# 内容简介

数字电路与系统是电子信息类专业的一门重要的专业基础平台课程。本书围绕该课程的各种概念、分析和设计方法，以通俗易懂的语言，对各部分内容的重点与难点进行了系统的描述和总结；针对不同的题型，给出了详尽的分析和解答。

本书是作者长期从事该课程教学的经验总结和多年的实际教学应用，内容充实，精练而不繁琐，浅显而不深奥，对各种常见问题有独到的见解和解释。全书共分为十二章，按国家教育部颁布的教学大纲顺序覆盖了该课程的全部内容。本书适用于参加该课程学习的初学者和参加考研的学子，也可供从事该课程教学的教师参考。

# 前 言

数字电路是IT大厦的基石，随着数字逻辑器件的诞生以及数字电路设计理论与方法的成熟，出现了诸如CPU、存储器等系列大规模数字集成电路，从而构成了数字系统的硬件基础，为系统软件和应用软件运行提供了强有力的物质保障。由数字电路构成的应用系统称为数字系统，人们日常使用的手机、计算机、MP3、DVD等均属数字系统所研究的范畴。目前，数字系统已广泛应用于通信、雷达、测试、工业控制等领域，进入了千家万户。数字电路与系统主要研究数字系统的硬件构成，硬件工程师与软件工程师的区别在于：前者更致力于打造完美汽车，后者更像是位赛车手。

由于数字电路与系统课程的内容庞杂，涉及大量的概念和电路分析设计方法，初学者很难对相关概念准确把握，解题面临较大难度。目前，该课程亦被全国许多高校列为电子信息类研究生的专业入学考试科目。很多考生在复习时发现，对一些基本概念已经遗忘，对课程内容难以形成一个清晰的脉络，解题技巧亟待提高。此外，对部分正欲承担该课程教学工作的青年教师来说，由于之前未从事过数字工程设计及科研工作，缺乏相应的工程应用背景经验，难以将教学过程简单化、通俗化，从而影响了课程教学效果。本书的目的正基于此，希望给广大的初学者、考研同学和青年教师们提供一位良师益友。

数字电路与系统课程的主要教学内容集中在逻辑代数、组合逻辑电路、同步时序逻辑电路和脉冲电路这四个部分。为适应各高校的具体要求，本书内容将尽可能涵盖大部分高校现行教材所涉猎的内容。每一章由本章概要、重点与难点、知识结构图、知识点详解、典型例题和习题详解六个小节组成。其中，本章概要构成每一章的第一大节，力争以通俗的语言介绍本章学习的目的和意义，理清概念间的脉络关系，介绍该部分知识的工程应用情况及发展方向。重点与难点、知识结构图和知识点详解三个小节构成每一章的第二大节，主要对教材学习内容和设计分析方法进行直观的总结，以便于系统复习与记忆。重点与难点部分还指出了每个知识点与典型例题或习题详解的对应关系，使读者复习与解题时能有的放矢。典型例题和习题详解两个小节构成每一章的第三大节，是每一章的重点。典

型例题大多选自研究生近年入学考题，习题详解则采自我校本科教学现行教材。书中对每道例题和习题进行了较深入全面的分析，解题技巧往往包含在每一道习题的“分析”部分中，许多最新的分析设计方法和作者十多年来的教学经验及成果也通过这种形式予以体现，这对现行教材内容来说是有力的补充。

全书共分为12章。第1、2章分别介绍了数字逻辑基础、逻辑门电路，是整个数字电路的基础；第3、4、5、6章分别介绍了组合逻辑电路、集成触发器、同步时序逻辑电路、异步时序电路，并对各种数字电路的分析与设计方法进行了总结；第7、8、9、10章分别介绍了集成存储器、可编程逻辑器件（PLD）、硬件描述语言（VHDL）和数字系统，这为研究通用或专用数字系统、超大规模片上集成系统打下了必要的基础；第11、12章分别介绍了脉冲波形的产生、整形以及数/模和模/数转换的工作原理及分析设计方法；本书最后还给出了两套模拟考试题及其解答。为方便读者查找，在附录1中列出了常用的TTL和CMOS系列集成电路功能索引；在附录2中介绍了EWB软件的使用方法，读者可利用该软件对本书的大部分习题进行仿真分析和验证。第1、2、3、4、5、9、11章是本课程的重点。正式出版前，本书已经在本科课程教学中试用三年，效果良好。

本书编写组由赵刚、杨鹏、冯进华、何培宇四位教师组成，编写工作由参加编写的教师集体讨论、分工编写、交叉修改、历经四年而完成。由赵刚担任主编，负责大纲拟定、组织编写与统稿工作。

在本书编写过程中，得到了四川大学电子信息学院数字电路与系统课程组全体同仁的大力支持，夏秀渝、陈瑛等老师也参与了本书部分内容的编写，周瑞东、张翀、江勇、孙立风、刘宗斌、张亮亮、赵烽琪、张翔、覃宇、王丰华、窦蕴蒲、何江波、汤智杰等同学积极地参与了本书录入校对工作，借本书出版之际，向他们表示衷心的感谢。

在本书出版过程中，得到了四川大学教务处和四川大学出版社的热情关怀，得到了周树琴老师的大量帮助，在此表示诚挚的谢意。

本书编写时，虽然立足通俗性和实用性，追求概念及理论脉络的直观清晰、解题方法的简捷与新颖，但由于编者水平有限，书中难免有不妥和错误之处，恳请读者批评指正。

编著者

2007年12月31日于四川大学

# 目 录

# 第 1 章　数字逻辑基础

## 本章概要

数是人类认识世界的一个有力武器，人的十个手指让十进制理所当然地成为了自然计数的基准。对一个数字系统来说，具有良好的运行可靠性尤为重要。在实际的数字电路实现中，二进制数字电路结构最为简单，并具有抗干扰能力强等特点，从而在数字系统中得到广泛应用。

与十进制相比，二进制数字位数更长，运算繁琐，但这可通过提高电路的工作速度予以弥补。由于八进制或十六进制与二进制之间转换方便，因而常常在日常编程中用其来减少数字的记录位数。与“数”具有可运算性不同，“码”则只是一种对数符和文字的表现形式，不是用来计数或参加运算的。对于二进制数人们并不习惯，为便于人机交互，数字系统在输出显示时往往已对二进制数进行了 BCD 码转换，BCD 码即是用 4 位二进制“码”来对 10 个十进制数进行的一种代码表示。

将所有的输入输出关系列举出来，则形成了真值表，它反映了所有输入输出之间的逻辑关系。与普通代数一样，可使用布尔代数形式来表达实际的逻辑变量之间的依存关系，从而为数字电路与系统的设计和分析提供了强有力的数学工具。对逻辑函数的化简将意味着对电子元器件的节省，美国工程师卡诺为此发明了“卡诺图”工具，使得对逻辑函数的化简变得直观简便。

## 重点与难点

数字逻辑是数字电路分析和综合的数学工具。通过本章的学习，首先应掌握数字系统中的常用数制、数制转换方法以及数码和常用编码，掌握逻辑代数的各种基本运算和逻辑代数的公理、定理及规则，并掌握几种导出逻辑门；然后重点掌握逻辑函数的表示方法和化简方法。

本章的难点是：

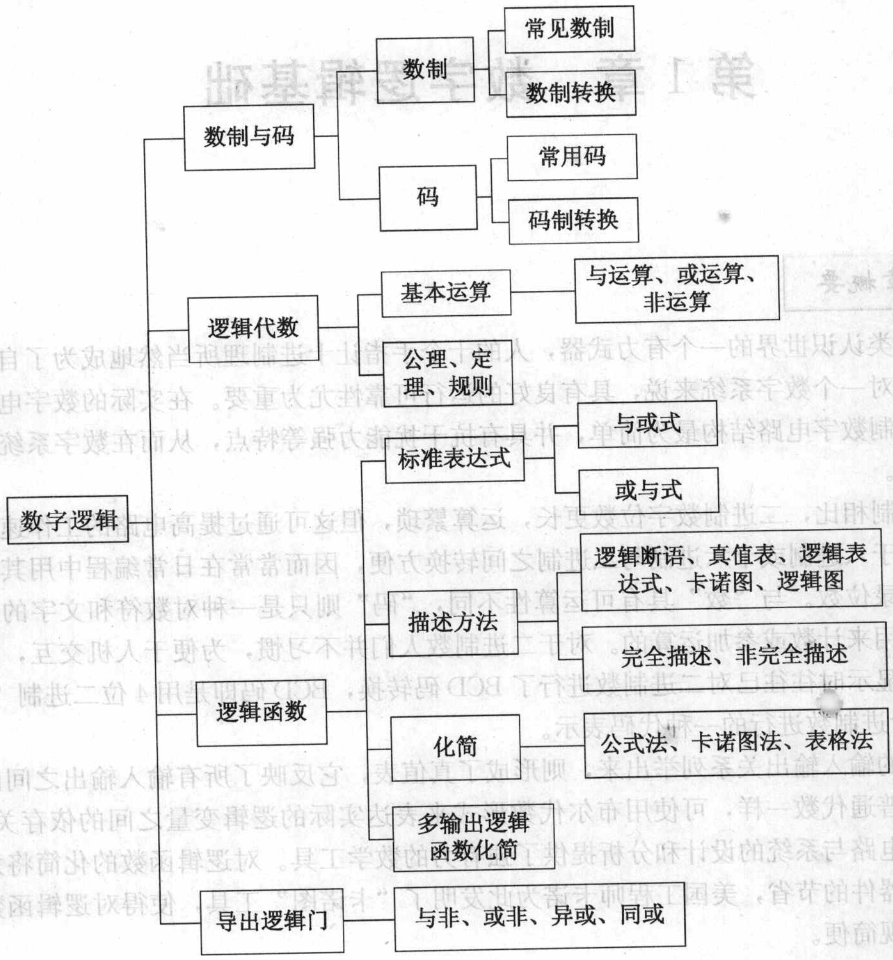
(1) 数制与码的相互转换（见“典型例题”的第 1 题和“习题详解”的第 1、2、3、4、5、6 题）。

(2) 反演律、添加项定理等定律的应用（见“典型例题”的第 2、3 题和“习题详解”的第 9、10、11、12 题）。

(3) 最大项与最小项的关系（见“习题详解”的第 16、17 题）。

(4) 卡诺图化简逻辑函数、任意项的处理（见“典型例题”的第 7 题和“习题详解”的第 14、15、16、17、18、20、21 题）。

## 知识结构图




## 知识点详解

### 1.1 数制与码

#### 1.1.1 数制

数制即计数的方法，常分为“有权”和“无权”两种。众所周知的十进制计数是有权计数的典型，而无权计数则较少，例如古希腊和罗马通用的数字（Ⅰ，Ⅱ，Ⅲ，…）。两者的不同之处在于：前者任一位数码代表的值不只是取决于数码本身，还取决于它所在的位置；后者数码表示的值只取决于数码本身，而与它所在的位置无关。

在有权计数中，任何一个数都可以由数码和一个小数点来表示（当然，我们往往还需要用符号位来表示数的正负）。一般来说，$R$ 进制的计数方法可表示为：$R$ 进制计数总共有 $R$ 个数码，0，1，2，…，$R-1$ 和一个小数点“.”。其计数规则以 $R$ 为基数，每计满 $R$ 向高位进一，即逢 $R$ 进一。其系数序列表达式为

$$(N)_R=(a_{n-1}a_{n-2}\cdots a_1a_0a_{-1}a_{-2}\cdots a_{-m})_R$$

其按权展开式为

$$(N)_R=\sum_{i=-m}^{n-1}a_iR^i\text{，}0\leqslant a_i\leqslant R-1$$

其中，$a_i$ 为各位的系数，$R^i$ 为各位的权。

在数字电路中，经常使用的有权计数有十进制数、二进制数、八进制数和十六进制数。

(1) 十进制数：以10为基数，共有10个数码，分别为0，1，2，3，4，5，6，7，8，9，每计满10向高位进一，其按权展开式为

$$(N)_{10}=\sum_{i=-m}^{n-1}a_i10^i\text{，}0\leqslant a_i\leqslant 9$$

(2) 二进制数：以2为基数，只有2个数码，分别为0和1，每计满2向高位进一，其按权展开式为

$$(N)_2=\sum_{i=-m}^{n-1}a_i2^i\text{，}0\leqslant a_i\leqslant 1$$

(3) 八进制数：以8为基数，共有8个数码，分别为0，1，2，3，4，5，6，7，每计满8向高位进一，其按权展开式为

$$(N)_8=\sum_{i=-m}^{n-1}a_i8^i\text{，}0\leqslant a_i\leqslant 7$$

(4) 十六进制数：以16为基数，共有16个数码，分别为0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F，每计满16向高位进一，其按权展开式为

$$(N)_{16}=\sum_{i=-m}^{n-1}a_i16^i\text{，}0\leqslant a_i\leqslant 15$$

对于不同进制数之间的转换，常用的方法有以下两种：

(1) 替换法：若要将 $R$ 进制数转换成 $S$ 进制数，只需将 $R$ 进制数按权展开，并将展开式中的基数数码（0，1，2，…，$R-1$）、进位码（$R$）以及指数全部替换为 $S$ 进制数，再按照 $S$ 进制数的运算规则运算，即可得到 $S$ 进制数。用公式可以表示为

$$(N)_R=\sum_{i=-m}^{n-1}(a_i)R^{(i)_R}=\sum_{i=-m}^{n-1}(a_i)S^{(i)_S}=(N)_S$$

其中，$(i)_R$ 和 $(i)_S$ 分别为 $i$ 的 $R$ 进制和 $S$ 进制表示。例如，要将十进制数139表示为8进制，则有：

$$\begin{aligned}(139)_{10}&=1\times10^2+3\times10^1+9\times10^0\\&=1\times(12)^2+3\times(12)^1+(11)\times(12)^0\\&=1\times144+3\times12+11\\&=144+36+11\\&=(213)_8\end{aligned}$$

实际上，各种数制的按权展开式都可以认为是不同数制到十进制数的转换公式。

(2) 基数连除（整数）、连乘（小数）法：将整数部分和小数部分分别处理。对于整数部分，用要转换的数除以新的进制基数，直到商为零，然后将所有余数按照先后顺序从低位到高位排列即得到转换的结果，这种方法可以简记为"连除取余反读出"；对于小数

部分，用要转换的数乘以新的进制基数，取出整数部分，直到小数部分为零（或者满足转换精度的要求），然后将整数得到的整数按照先后顺序从高位到低位排列即得到转换的结果，这种方法可以简记为“连乘取整顺读出”。

从 $R$ 进制数到 $S$ 进制数的转换过程中，若 $R$ 小于 $S$，则运用第一种方法较为简单，而若 $R$ 大于 $S$，则运用第二种方法较为简单。

特别的是，对于二进制与八进制和十六进制的转换，八进制的基数是 8，正好是 $2^3$，这说明，每一个八进制数都可用三位二进制数来表示，因此，二进制转换为八进制时，只需将二进制数按照三位一组来分组，然后分别将其转换成一位八进制数即可；反过来，八进制转换成二进制时，只需将每一位八进制数转换成相应的三位二进制数即可。二进制数与十六进制数的转换与此类似，因为 $2^4=16$，所以每一位十六进制数正好对应 4 位二进制数。

例如，将二进制数 $M$ 转换成八进制数 $N$，则有：

$$
\begin{aligned}
(M)_2 &= a_{3m+2}2^{3m+2}+a_{3m+1}2^{3m+1}+a_{3m}2^{3m}+a_{3m-1}2^{3m-1} \\
&\quad +\cdots+a_5 2^5+a_4 2^4+a_3 2^3+a_2 2^2+a_1 2^1+a_0 2^0 \\
&=(a_{3m+2}2^2+a_{3m+1}2^1+a_{3m}2^0)(2^3)^m+\cdots \\
&\quad +(a_5 2^2+a_4 2^1+a_3 2^0)(2^3)^1+(a_2 2^2+a_1 2^1+a_0 2^0)(2^3)^0 \\
&=b_m 8^m+\cdots+b_1 8^1+b_0 8^0 \\
&=(N)_8
\end{aligned}
$$

其中，八进制数 $N$ 的每一项正好都是三位二进制数的组合。

### 1.1.2 码

码与数是不同的，通常所说的数是可以参加运算的数值. 而码是以二进制形式出现的对数符或者文字的编码。常见的码可以分成“有权”码和“无权”码。实际应用中，常用到的码有自然二进制码、ASCII 码、各种 BCD 码等。在各种 BCD 码中，最常用的是 8421BCD 码，它是用二进制对一位十进制数进行的一种编码，主要是为了数字系统中的数码管显示。常见的各种 BCD 码如表 1−1 所示。

表 1−1　常用 BCD 码

| 十进制数 | 有权码 | | | | 无权码 | | |
|---|---|---|---|---|---|---|---|
| | 8421 | 4221 | 2421 | 5421 | 余 3 码 | 格雷码 | 步进码 |
| 0 | 0000 | 0000 | 0000 | 0000 | 0011 | 0000 | 00000 |
| 1 | 0001 | 0001 | 0001 | 0001 | 0100 | 0001 | 10000 |
| 2 | 0010 | 0010 | 0010 | 0010 | 0101 | 0011 | 11000 |
| 3 | 0011 | 0011 | 0011 | 0011 | 0110 | 0010 | 11100 |
| 4 | 0100 | 1000 | 0100 | 0100 | 0111 | 0110 | 11110 |
| 5 | 0101 | 0111 | 1011 | 1000 | 1000 | 0111 | 11111 |
| 6 | 0110 | 1100 | 1100 | 1001 | 1001 | 0101 | 01111 |
| 7 | 0111 | 1101 | 1101 | 1010 | 1010 | 0100 | 00111 |
| 8 | 1000 | 1110 | 1110 | 1011 | 1011 | 1100 | 00011 |
| 9 | 1001 | 1111 | 1111 | 1100 | 1100 | 1101 | 00001 |

另外，在数字系统中，数的正负也是数值化了的。我们规定：正数的最高位为 0，负

数的最高位为1。考虑到数的正负位，在数字系统中，带符号位的数有三种表示方法，即原码表示法、反码表示法和补码表示法。补码的特殊性在于：它对0的表示是唯一的，这样在运算或者其他处理的时候不会产生错误，而原码和反码对0都有两种表示。

## 1.2 逻辑代数

### 1.2.1 逻辑代数

逻辑代数又称布尔代数或开关代数，即二进制数的逻辑运算，只有两个数字，分别为0和1，代表两种相反的状态，表示运算结果为真或者为假。与二进制数的算术运算不同的是，在逻辑运算中，只有三种运算（与、或、非)，任何两种都可以构成封闭的运算集。利用这三种运算，可以得到各种复合逻辑，并可以得到用以实现这些逻辑运算的各种导出逻辑门。

### 1.2.2 逻辑代数的基本公理、定理和规则

(1) 逻辑代数的基本公理如表1−2所示。

**表1−2 常用公理**

| | |
|---|---|
| $\overline{1}=0$ | $\overline{0}=1$ |
| $1\cdot1=1$ | $0+0=0$ |
| $1\cdot0=0$ | $0+1=1$ |
| $0\cdot1=0$ | $1+0=1$ |
| $0\cdot0=0$ | $1+1=1$ |

(2) 基本定理及规则。

摩根定理：即反演律，其含义为：对于任何一个逻辑函数$F(A, B, C, \cdots)$，若将其中的“+”变换成“·”，“·”换成“+”，将“0”换成“1”，“1”换成“0”，再将原变量变为反变量，反变量变为原变量，并保持运算顺序不变，则得到的新的逻辑函数即为原函数$F$的反函数$\overline{F}(A, B, C, \cdots)$。

代入规则：在任何一个含有变量$A$的恒等式中，用函数$F$代替$A$，则恒等式仍然成立。

对偶规则：对于任何一个逻辑函数$F(A, B, C, \cdots)$，若将其中的“+”变换成“·”，“·”换成“+”，将“0”换成“1”，“1”换成“0”，并保持运算顺序不变，则得到的新的逻辑函数即为原函数$F$的对偶函数$F_d(A, B, C, \cdots)$。

(3) 常用公式。

逻辑代数中的常用公式如表1−3所示。

**表1−3 逻辑代数中的基本公式**

| $N$ | 名称 | 公式 | |
|---|---|---|---|
| 1 | 0−1律 | $A\cdot0=0$ | $A+1=1$ |
| 2 | 自等律 | $A\cdot1=A$ | $A+0=A$ |

续表 1－3

| $N$ | 名称 | 公式 | |
|---|---|---|---|
| 3 | 重叠律 | $A\cdot A=A$ | $A+A=A$ |
| 4 | 互补律 | $A\cdot\overline{A}=0$ | $A+\overline{A}=1$ |
| 5 | 交换律 | $A\cdot B=B\cdot A$ | $A+B=B+A$ |
| 6 | 结合律 | $A\cdot(B\cdot C)=(A\cdot B)\cdot C$ | $A+(B+C)=(A+B)+C$ |
| 7 | 分配律 | $A\cdot(B+C)=AB+AC$ | $A+BC=(A+B)(A+C)$ |
| 8 | 非非律 | $\overline{\overline{A}}=A$ | $\overline{\overline{A}}=A$ |
| 9 | 反演律 | $\overline{AB}=\overline{A}+\overline{B}$ | $\overline{A+B}=\overline{A}\cdot\overline{B}$ |
| 10 | 吸收律 | $A+AB=A$ | $A(A+B)=A$ |

几个重要定理：（1）合并律　$AB+A\overline{B}=A$；$(A+B)(A+\overline{B})=A$

（2）吸收律　$A+\overline{A}B=A+B$；$A(\overline{A}+B)=AB$

（3）冗余律　$AB+\overline{A}C=AB+\overline{A}C+BC$；

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

冗余律推论　$AB+\overline{A}C=AB+\overline{A}C+BCF$（$F$ 为任意变量）

## 1.3　逻辑函数的描述

对于任何一个逻辑函数，都可以由真值表化简为两种标准形式：积之和标准型与和之积标准型。

### 1.3.1　最小（大）项与真值表

我们把包含所有变量的乘积项称为函数 $F$ 的最小项，记为 $m_i$，其中每个变量出现且仅出现一次，可以是原变量，也可以是反变量，变量值为 1 时用原变量表示，为 0 时用反变量表示；同样，把包含所有自变量的和项称为函数的最大项，记为 $M_i$，其中每个变量出现且仅出现一次，可以是原变量，也可以是反变量，变量值为 1 时用反变量表示，为 0 时用原变量表示。对于真值表中的值，1 对应最小项，0 对应最大项，即用最小项表示的逻辑函数应包含所有值为 1 的项，用最大项表示的逻辑函数应包含所有值为 0 的项。

### 1.3.2　最小项和最大项的作用

函数展开为积之和标准型的意义在于，当某个最小项的值为 1 时，则函数为 1；函数展开为和之积标准型的意义在于，当某个最大项的值为 0 时，则函数为 0。最小项和最大项的关系可以描述为

$$m_i=\overline{M_i},\ m_i=(M_{n-1-i})_d$$

即对于逻辑函数的最大项和最小项，若其序号相同，则二者互补；若其序号互补（或者其变量相同），则二者互为对偶。其中，对于 $m$ 个逻辑变量的函数，$n=2^m$。

对于同一个逻辑函数，用最小项表示与用最大项表示是等价的。

例如，对于逻辑函数 $F(A,B)$，其真值表如下表所示：

| A | B | F |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

设 $F_1$、$F_2$ 分别为该逻辑函数的最小项表示和最大项表示，列出 $F_1$、$\overline{F_1}$、$F_2$ 的真值表，如下表所示：

| A | B | F | $F_1$ | $\overline{F_1}$ | $F_2$ |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 |

根据最小项和最大项的定义，有：

$$F_1 = \sum m(0,3) = \overline{A}\,\overline{B} + AB$$

$$\overline{F_1} = \sum m(1,2) = \overline{A}B + A\overline{B}$$

$$F_2 = \prod M(1,2) = (A+\overline{B})(\overline{A}+B)$$

将 $\overline{F_1}$ 取反，可得

$$\overline{\overline{F_1}} = \overline{\overline{A}B + A\overline{B}} = (A+\overline{B})(\overline{A}+B) = F_2$$

根据非非律，$\overline{\overline{F_1}} = F_1$，由此可得 $F_1 = F_2$。可见，用最小项和最大项表示的逻辑函数虽具有不同的形式，但是等价的。

### 1.3.3　完全描述和非完全描述

对于 $m$ 个变量的逻辑函数，对于 $2^m$ 组输入变量取值，函数（输出）都有确定的取值，这样的逻辑函数称为完全描述的逻辑函数。也就是说，对于任何完全描述的逻辑函数，其输出与所有 $2^m$ 个输入最小项都有关。

相反，若函数（输出）取值与其中一些输入最小项没有关系，或者某些输入是被客观禁止的，则称为非完全描述的逻辑函数，并且称这些最小项为无关最小项。无关最小项有两种情况：一种是不可能出现或者不允许出现的，常称为约束项；另一种是不影响输出的，常称为任意项或者随意项。在后面的章节中，常将这两种情况都作为随意项处理，以利于逻辑函数的化简。

## 1.4　逻辑函数化简

逻辑函数的标准表达式是唯一的，但一般不是最简的。逻辑函数最简表达式的准则是：

（1）逻辑函数的与或表达式（或与表达式）的与项（或项）最少；

（2）要求每一个与项（或项）的变量数最少。

常用的化简方法有：代数化简法、卡诺图化简法和表格法。

### 1.4.1 代数化简法

代数化简法，实际上就是运用逻辑代数的基本定理和规则对逻辑函数进行反复运算求得最简表达式的过程，其技巧性很强，化简的方法往往不是唯一的。常用的化简方法包括：合并项法、吸收法、消去法、配项法和添项消项法等。

### 1.4.2 卡诺图化简法

（1）卡诺图。将 $m$ 个变量的全部最小项各用一个小方块表示，并使具有逻辑相邻性的最小项在几何位置上也相邻，所得到的图形称为 $m$ 变量卡诺图。所谓逻辑相邻，即两个最小（大）项中只有一个变量不相同，且分别为原变量和反变量，而其他所有变量都相同。

（2）卡诺图化简。卡诺图化简逻辑函数的基本思想是：合并相邻最小（大）项，消去相邻最小（大）项中的互补变量，而其他相同的变量则保留。利用的基本原理是公式 $A+\overline{A}=1$（互补律），相邻最小（大）项的合并规律按 $2^k$（$k=0$，1，…，$n$）进行的。当 $k=1$ 时，可以消去一个变量；当 $k=2$ 时，可以消去两个变量……以此类推，当 $k=n$ 时，可以消去 $n$ 个变量，即合并后的表达式为常量 1（最小项和最大项均为 0），其物理意义是函数包含全部最小（大）项且永远为 1（0）。

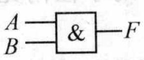
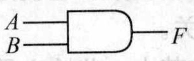
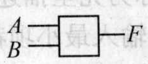
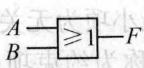
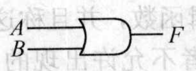
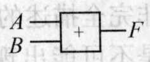
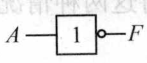
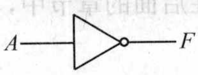
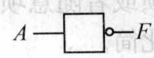
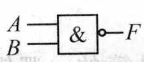
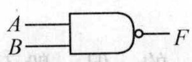
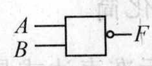
### 1.4.3 表格法

表格法的基本原理仍然是从合并两个相邻最小项出发，逐步化简，最终得到最简表达式。与卡诺图不同的是，表格法在化简的过程中有严格的算法，便于编制计算机解题程序，由计算机完成函数化简。这种方法适合于多变量、多输出函数的化简。

## 1.5 常见逻辑门

逻辑门是逻辑功能的符号表示，常见逻辑门见表 1－4。

**表 1－4 常见逻辑门**

| 名称 | 国家标准符号 | 国外符号 | 常见符号 | 逻辑关系 |
|---|---|---|---|---|
| 与门 |  |  |  | $F=A\cdot B$ |
| 或门 |  |  |  | $F=A+B$ |
| 非门 |  |  |  | $F=\overline{A}$ |
| 与非门 |  |  |  | $F=\overline{AB}$ |