SHUZIDIANZIJISHU

# 数字电子技术

程 开 明 主编

重庆大学电子学教研组编

SZDZJS

重 庆 大 学 出 版 社

# 数 字 电 子 技 术

重庆大学电子学教研组编

程 开 明 主编

重 庆 大 学 出 版 社

## 内 容 简 介

本书按照国家教委批准的《高等工业学校电子技术基础课程教学基本要求》编写。内容有：逻辑代数基础、逻辑门电路、组合逻辑电路、触发器、时序逻辑电路、集成存储器、脉冲的产生和整形、数模·模数转换器、数字电路读图练习，各章均附有习题。可供60～70学时课堂教学使用。

本书可作高等学校电气类、电子类和其它相近专业的教材，也可供有关工程技术人员参考。

**数字电子技术**

程开明 主编

责任编辑 谭 敏

·

重庆大学出版社出版发行

新 华 书 店 经 销

重庆建筑工程学院印刷厂印刷

·

开本：787×1092 1/16 印张：13.25 字数： 331 千

1992 年 5 月第 1 版 1992 年 5 月第 1 次印刷

印数： 1-5000

标准书号：ISBN 7-5624-0443-7 / TN·10 定价：3.87元

# 初 版 序

根据电气类电子技术基础课教学需要，我们采用从模拟到数字的体系，编写了《模拟电子技术》、《数字电子技术》讲义。经多年教学实践的充实，最后按照1987年国家教委批准的《高等工业学校电子技术基础课程教学基本要求》进行了修订，本书是数字电子技术基础部分。

本书以基本概念、基本知识、基本方法为主线，较系统地介绍了数字电子技术基础的逻辑单元、典型电路及应用。单元电路既保留了各种功能双极型的内容，又增加了CMOS电路的比重。小规模集成电路的逻辑符号一律采用国家标准。

为了适应数字电子技术日新月异的发展，在介绍小规模集成电路基础上，增加了中、大规模集成电路的内容。中、大规模电路的应用实例，在相应的章节分别作了归纳。

本书在内容编排、方法介绍上，力求深入浅出，循序渐进，以点带面，重点突出。

除基本内容外，书中有的章节还有拓宽部分，均以＊号标出，可供选用。

本书前六章由程开明执笔，第七、九章及前七章习题由唐治德执笔，第八章由李琦执笔，邓晓琳作了部分底图的绘制。程开明任主编，负责组织和定稿，唐治德协助主编工作。在编写过程中，郑荣义、周宝薇、谭金蓉、杨永明、曾孝平等同志参加了部分工作，给予了很大支持。

书稿承许德沛教授主审，杨永臻教授、李时光教授审阅了讲义。

对为本书进行审阅并提出宝贵意见，以及在编写、出版过程中给予帮助和支持的同志，在此，一并表示诚挚的谢意。

由于我们水平有限，虽然根据各方意见多次修改，书中一定还有不少缺点、错误，恳请读者批评、指正。

编者

1991年 3 月于重大

# 目　录

# 第一章　逻辑代数基础

逻辑代数又叫做开关代数，是19世纪一位英国数学家布尔（G.Bool）创立的，因而又名布尔代数。它是研究数字电路的数学工具，为分析和设计数字电路提供了理论基础。本章重点介绍化简逻辑函数的代数法及图形法。

## §1.1　概　　述

电子电路中的电信号分两大类。一类，在时间上、数值上的变化是连续的、平滑的信号，叫做模拟信号。例如，从热电偶得到的电压信号，就是一个模拟信号。另一类，在时间上、数值上的变化是离散的（不连续的）信号，叫做数字信号。例如，生产自动线上产品零件的个数，就是一个数字信号。处理模拟信号的电路，叫做模拟电路；处理数字信号的电路，叫做数字电路。

数字电路研究的对象，不是输出与输入之间的数量关系，而是它们的逻辑关系。所用的方法是逻辑代数。

数字电路中的单元电路，为各种各样的开关。单元电路叫做开关电路或开关器件。由这些开关器件，运用开关理论，采用不同的连接方式，就可构成各种功能的数字电路或数字系统（例如数字计算机）。

数字电路的优点：

1. 精度高。可用增加数字信号的位数来达到所需精度的要求；

2. 稳定可靠。只需区分信号的有无，因而抗干扰性能强；

3. 有处理本领。可以对信号进行存储和判断；

4. 通用性强。可用标准化部件构成各种电路。

所以，随着大规模、超大规模数字集成电路的出现，以及计算技术的广泛采用，数字电路越来越普及。

### 1.1.1　数制和码制

**一、数制**

数制即计数体制。各种计数制都有两个基本特点：一是所用数码（或数符）的个数（叫做基数）；二是位权，它表明不同位数的大小。常用的数制有：

1. 10进制　日常生活中，最常用的是10进制，其特点：

(1) 有0～9十个数符，基数是10。

(2) 位权为$10^i$。基数的$i$次幂叫做第$i$位的位权，故低位数与相邻高位数的关系为逢十进一。

例1-1　$123 = 1\times10^2 + 2\times10^1 + 3\times10^0$

2. 2进制　数字电路中常用2进制，其特点：

(1) 有0、1两个数符，基数为2。

(2) 位权为 $2^i$。低位数与相邻高位数的关系为逢二进一。

3. 8进制 书写计算机程序时常用8进制、16进制。8进制的特点为:

(1) 有0～7八个数符。基数为8。

(2) 位权为 $8^i$，故逢八进一。

4. 16进制 16进制的特点为:

(1) 有0～9、$A$(10)、$B$(11)、$C$(12)、$D$(13)、$E$(14)、$F$(15)等16个数符，基数为16。

(2) 位权为 $16^i$，逢16进一。

由上面分析可知，以$N$为基数的计数体制，叫做$N$进制。凡$N \neq 10$的$N$进制数$D$所对应的10进制值，可按下式展开

$$D = \sum K_i \times N^i$$

即所谓按权展开再相加。其中$K_i$为第$i$位的系数，可为基数中的任一个数符。

例1-2 $(1101)_2 = 1\times 2^3 + 1\times 2^2 + 0\times 2^1 + 1\times 2^0 = (13)_{10}$

$(54)_8 = 5\times 8^1 + 4\times 8^0 = (44)_{10}$

$(4E)_{16} = 4\times 16^1 + 14\times 16^0 = (78)_{10}$

**表1.1.1 常用2-10进制（BCD）编码表**

| 编码种类 / 10进制数 | 8421码 | 余三码 | 2421($A$)码 | 2421($B$)码 | 5211 | 余三循环码 | 右移码 |
|---|---|---|---|---|---|---|---|
| 0 | 0000 | 0011 | 0000 | 0000 | 0000 | 0010 | 00000 |
| 1 | 0001 | 0100 | 0001 | 0001 | 0001 | 0110 | 10000 |
| 2 | 0010 | 0101 | 0010 | 0010 | 0100 | 0111 | 11000 |
| 3 | 0011 | 0110 | 0011 | 0011 | 0101 | 0101 | 11100 |
| 4 | 0100 | 0111 | 0100 | 0100 | 0111 | 0100 | 11110 |
| 5 | 0101 | 1000 | 0101 | 1011 | 1000 | 1100 | 11111 |
| 6 | 0110 | 1001 | 0110 | 1100 | 1001 | 1101 | 01111 |
| 7 | 0111 | 1010 | 0111 | 1101 | 1100 | 1111 | 00111 |
| 8 | 1000 | 1011 | 1110 | 1110 | 1101 | 1110 | 00011 |
| 9 | 1001 | 1100 | 1111 | 1111 | 1111 | 1010 | 00001 |

**表1.1.2 4位循环码**

| 10位进数 | 循环码 | 10进制数 | 循环码 |
|---|---|---|---|
| 0 | 0000 | 8 | 1100 |
| 1 | 0001 | 9 | 1101 |
| 2 | 0011 | 10 | 1111 |
| 3 | 0010 | 11 | 1110 |
| 4 | 0110 | 12 | 1010 |
| 5 | 0111 | 13 | 1011 |
| 6 | 0101 | 14 | 1001 |
| 7 | 0100 | 15 | 1000 |

$N$进制之间相互转换的方法，计算机原理课中会详细介绍。

**二、码制**

数码不仅可以表示数量的大小，还可用来表示不同的事物，此时的数码叫做代码，为不同事物的代号。例如招生考号1101与1011仅代表不同的考生，失去了数量大小的含义。

为了便于记忆和查找，编制代码时所遵循的规则，即编码方案，叫做码制。例如，

用 2 进制代码表示一位10进制数，要用 4 位代码，这种编码叫做 2 -10 进制编码，简称BCD①码。由于 4 位 2 进制代码共有$2^4=16$种组合，如果任取其中的10个组合，并按不同的顺序排列，就可得许多不同的编码，表1.1.1为常用的2-10进制编码表。表1.1.2的码制，叫做循环码或格雷码。其中，8421码、2421码、5211码为恒权代码，其余为无权代码。例如在8421码中，如果把每一个代码都看作一个 4 位 2 进制数，其权依次是8421，那么，这个代码的数值恰好等于它所代表的10进制数的大小。

### 1.1.2 算术运算和逻辑运算

在数字电路中，2 进制数码不仅可以表示数量的大小，还可表示事物的状态。当两个 2 进制数码表示两个数量大小时，它们之间可进行数值运算，即算术运算。2 进制的算术运算和10进制的算术运算基本相同，区别是两相邻进位关系为逢二进一。

当两个 2 进制数码表示不同逻辑状态时，它们之间的因果关系可进行逻辑运算。算术运算与逻辑运算有本质的差别，下面重点介绍逻辑运算的各种规则。

## §1.2 逻辑函数

### 1.2.1 几个基本概念

**一、逻辑状态表示法**

在日常生活中，存在着大量的两种完全对立的逻辑状态，例如：

一 种 状 态　高电位　有信号　真　是　…

另一种状态　低电位　无信号　假　非　…

两种对立的状态，在数字电路中，用逻辑符号“1”和“0”表示。这里的“1”和“0”并不表示数量的具体大小，有别于数学中的 1 和 0，而只是作为一种符号，称为逻辑 1 和逻辑 0。

**二、两种逻辑体制**

两种对立的逻辑状态用 1 和 0 表示，有两种方法，即两种逻辑体制。用 1 表示高电位，用 0 表示低电位；反之，也可用 0 表示高电位，用 1 表示低电位。前种表示方法叫做正逻辑体制，简称正逻辑；后一种表示方法叫做负逻辑体制，简称负逻辑。今后，如无特殊声明，皆为正逻辑。

**三、高、低电平的规定**

在逻辑电路中，电位常用电平一词来描述，高电位即高电平，低电位即低电平。本来电位和电平是有区别的，电位指具体的数量大小，电平则是指的范围，但在数字电路中不加区别而混用。

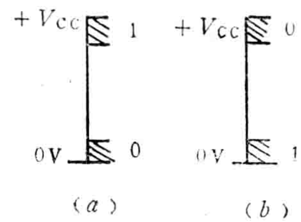



图1.2.1 正负逻辑
(a)正逻辑 (b)负逻辑

由于温度变化、电源电压波动，元件特性变化以及干扰等原因，实际的高、低电平都不是一个固定的值，通常考虑一个变化范围，如图 1.2.1 所示。如果在此范围内，就判断

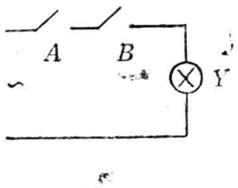
①BCD 是 Bjnary-Cooled-Decjmal 的缩写

为1或0。并且把高电平的下限值，规定为标准高电平 $V_{SH}$，把低电平的上限值，规定为标准低电平 $V_{SL}$。高电平过低，或低电平过高，都会破坏正常的逻辑关系，因此在逻辑系统中，应保证实际高电平≥$V_{SH}$，实际低电平≤$V_{SL}$。

### 1.2.2 三种基本逻辑关系

基本逻辑关系有逻辑“与”、逻辑“或”和逻辑“非”三种。

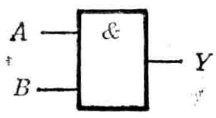
1. 逻辑“与” 在图1.2.2(*a*)照明电路中，以开关合上为条件，灯亮作结果，则开关 *A*、*B* 都合上（条件全部具备时），灯 *Y* 才亮（结果才会发生）。就是说，出现某种结果的条件全部具备时，结果才会发生，这种条件与结果的关系，叫做逻辑与。




(*a*)

| *A* | *B* | *Y* |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

(*b*)




(*c*)

图1.2.2 与逻辑关系

(*a*)电路 (*b*)真值表 (*c*)逻辑符号

采用正逻辑，条件具备（开关合上）用1表示，条件不具备（开关打开）用0表示，灯亮用1表示，灯灭用0表示。则图*a*的与逻辑关系可用图(*b*)表格表示。表中 *A*、*B* 表示逻辑条件，叫做输入逻辑变量，*Y* 作为逻辑结果，它依赖于逻辑变量，叫做逻辑函数。由于每个变量有两种取值，*n* 个变量则有 $2^n$ 种取值组合。本例为两个变量，故为 $2^2=4$ 种组合。这种完整地表达自变量所有可能取值组合与相应函数值关系的表格，叫做真值表。

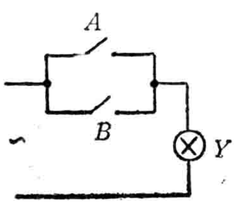
由真值表可见，只有当逻辑变量 *A*、*B* 全为1时，逻辑函数*Y*才为1。这种关系和算术中的乘法相似，所以，逻辑与又叫做逻辑乘。其表达式为

$$Y=A\cdot B=AB$$

式中“·”表示*A*与*B*逻辑相乘，为了书写简便“·”常常省去不写。

图(*c*)为与逻辑关系的逻辑符号。

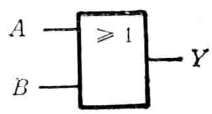
2. 逻辑“或” 在图1.2.3中，以开关合上为条件，灯亮作结果，则开关*A*或*B*或*A*及*B*合上（只要一个以上条件具备时），灯 *Y* 就会亮（结果就会发生）。就是说，出现某种




(*a*)

| *A* | *B* | *Y* |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

(*b*)




(*c*)

图1.2.3 或逻辑关系

(*a*)电路 (*b*)真值表 (*c*)逻辑符号

结果的条件中，只要一个以上具备时，结果就会发生，这种条件与结果的关系。叫做逻辑或。

采用正逻辑，得真值表($b$)，由真值表可见，逻辑变量$A$、$B$之中，只要有一个是1，逻辑函数$Y$就为1。这种关系和算术中的加法相似，因此就把逻辑或叫做逻辑加。其表达式为：

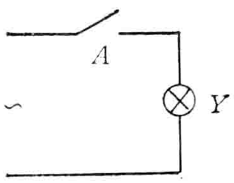
$$Y = A + B$$

式中“+”表示逻辑相加而不是算术相加。这里的1，0是表示两种不同逻辑状态的符号，没有数量的意思。例如$A$为高电平，$B$为高电平，那么在或逻辑关系中，$Y$也为高电平。所以有1 + 1 = 1。

图($c$)为或逻辑关系的逻辑符号。

上面，采用正逻辑，图1.2.3($a$)为或逻辑。相反，如果采用负逻辑，就得图1.2.2($b$)的真值表，为与逻辑。同理，图1.2.2($a$)采用正逻辑为逻辑与，采用负逻辑就为逻辑或。由此可知，逻辑关系与逻辑体制有关。一个电路（一个系统），只有当逻辑体制一定时，逻辑的性质才是确定的。

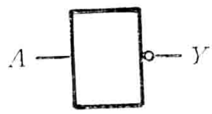
3. 逻辑“非”（反） 在图1.2.4中，以开关打开为条件，灯亮作结果，则开关$A$打开（条件具备时），灯$Y$不亮（结果不发生）；相反，开关$A$合上（条件不具备时），灯$Y$亮（结果必然发生），这种条件与结果的关系，叫做逻辑非或逻辑反。




($a$)

| $A$ | $Y$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

($b$)




($c$)

图1.2.4 非逻辑关系
($a$)电路 ($b$)真值表 ($c$)逻辑符号

采用正逻辑，得图($b$)真值表，图($c$)为非逻辑关系的逻辑符号。函数表达式为

$$Y = \overline{A}$$

## 1.2.3 复合逻辑运算

三种最基本的逻辑关系，也是三种最基本的逻辑运算，分别叫做与运算，简称乘法运算；或运算，简称加法运算；非运算，简称求反运算。此外，还经常用到与非、或非、与或非、异或、同或等运算，它们都是由这三种基本逻辑运算组合而成的复合运算。

**一、与非运算** $Y = \overline{AB}$

逻辑符号如图1.2.5($a$)，图上小圆圈表示非运算。可见，它是由与运算及求反运算组合而成，先与后求反。

**二、或非运算** $Y = \overline{A + B}$

逻辑符号如图1.2.5($b$)。它是先作或运算，再作非运算。

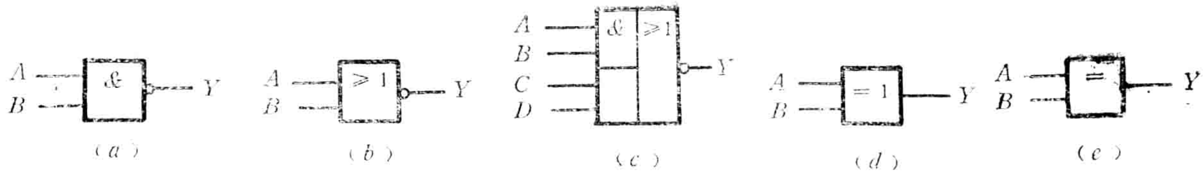



图 1.2.5 复合逻辑关系

**三、与或非运算** $Y=\overline{AB+CD}$

逻辑符号如图1.2.5($c$)，它是先与后或再求反。

**四、异或运算** $Y=A\overline{B}+\overline{A}B=A\oplus B$

两输入变量相同时，输出为 0；两输入变量相异时，输出为 1，这种逻辑关系，叫做异或，简记为$Y=A\oplus B$，读作$A$异或$B$。逻辑符号如图1.2.5($d$)。

**五、同或运算** $Y=\overline{A}\,\overline{B}+AB=A\odot B$

和异或正好相反，两输入变量相异时，输出为 0；两输入变量相同时，输出为 1，这种逻辑关系，叫做同或，简记为$Y=A\odot B$，读作$A$同或$B$。逻辑符号如图1.2.5($e$)。它们仍由三种最基本的运算复合而成。

## §1.3 逻辑代数的基本定律

### 1.3.1 定理和恒等式

**一、定　理**

| | | | |
|---|---|---|---|
| 定理 1 | 自等律 | $A+0=A$ | $A\cdot 1=A$ |
| 定理 2 | 0－1律 | $A+1=1$ | $A\cdot 0=0$ |
| 定理 3 | 重叠律 | $A+A=A$ | $A\cdot A=A$ |
| 定理 4 | 互补律 | $A+\overline{A}=1$ | $A\cdot\overline{A}=0$ |
| 定理 5 | 吸收律 | $A+AB=A$ | $A(A+B)=A$ |
| 定理 6 | 非非律 | $\overline{\overline{A}}=A$ | |
| 定理 7 | 交换律 | $A+B=B+A$ | $AB=BA$ |
| 定理 8 | 结合律 | $(A+B)+C=A+(B+C)$ | $(AB)C=A(BC)$ |
| 定理 9 | 分配律 | $A(B+C)=AB+AC$ | $A+BC=(A+B)(A+C)$ |
| 定理10 | 反演律（摩根定理） | $\overline{A+B}=\overline{A}\cdot\overline{B}$ | $\overline{AB}=\overline{A}+\overline{B}$ |

以上10个定理可借助于真值表，用 3 种基本逻辑运算证明。

例1-3　证明反演律 $\overline{AB}=\overline{A}+\overline{B}$。

将变量各种取值组合代入等式，进行计算的结果见表1.3.1。

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

## 二、常用恒等式

1. $AB+A\overline{B}=A$

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

此式叫做合并律。

2. $A+\overline{A}B=A+B$

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

表1.3.1

| $A$ | $B$ | $AB$ | $\overline{AB}$ | $\overline{A}$ | $\overline{B}$ | $\overline{A}+\overline{B}$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 |

此式说明，在一个与或表达式中，如果一项的反是另一项的因子，则此因子是多余的。故它是另一种形式的吸收律。

3. $AB+\overline{A}C+BC=AB+\overline{A}C$

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

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

推论 $AB+\overline{A}C+BCD=AB+\overline{A}C$

此式说明，在一个与或表达式中，如果两项分别包含 $A$ 和 $\overline{A}$，而其余的因子为第三项的因子，则第三项是多余的。此式又叫添加律。

4. $\overline{A\overline{B}+\overline{A}B}=AB+\overline{A}\overline{B}$，即 $\overline{A\oplus B}=A\odot B$

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

同理 $\overline{A\odot B}=A\oplus B$

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

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

公式说明，由两项组成的与或表达式中，如果一项含$A$，另一项含$\overline{A}$，那么将这两项其余部分各自求反，就得这个函数的反函数。

值得指出的是，定理和恒等式反映的是逻辑关系，不是数量之间的关系。由于逻辑代数中没有逻辑减法及逻辑除法，故初等代数中的移项规则（移加作减，移乘作除）这里不适用。

# 1.3.2 逻辑运算的基本规则

## 一、代入规则

在任何逻辑等式中，如果在所有地方出现的某一变量，都以一个逻辑函数代入，则等式仍然成立。这个规则叫做代入规则。

例1-4 $\overline{AB}=\overline{A}+\overline{B}$

式中 $B$ 以 $BC$ 代替，则

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

有了代入规则，上述公式不仅对变量适用，把逻辑函数看作变量，公式仍然成立，即公式由变量就扩大到函数了。

例1-5　$A(B+C) + A\overline{B+C} = A$

**二、反演规则**

任何一个逻辑函数$Y$中，如果将所有的“·”换成“+”，所有的“+”换成“·”；所有的“0”换成“1”，所有的“1”换成“0”；所有的原变量换成反变量，所有的反变量换成原变量，所得的新函数就是函数$Y$的反函数$\overline{Y}$，这就是反演规则。

运用反演规则时注意：不属于单个变量上的反号应该保留。

有了反演规则，为求反函数提供了方便。

例1-6　$Y = \overline{A + 0 + 1} = \overline{1} = 0$，则$\overline{Y} = \overline{\overline{A}\cdot 1\cdot 0} = \overline{0} = 1$

例1-7　$Y = A\overline{B} + \overline{CD}$，则 $\overline{Y} = \overline{A} + B(C+\overline{D})$

**三、对偶规则**

任何一个逻辑函数$Y$中，如果将所有的“·”换成“+”，所有的“+”换成“·”；所有的“0”换成“1”，所有的“1”换成“0”，所得的新函数就是函数$Y$的对偶式$Y'$。

当某逻辑等式成立，则其对偶式也成立，这就是对偶规则。

对照前面定理左右两边等式，不难发现，它们互为对偶式。

例1-8　$A(B+C) = AB + AC$　　有对偶式

$A + BC = (A+B)(A+C)$

有了对偶规则就使要证明、要记忆的公式减少了一半。正因为如此，常用恒等式就未给出对偶式了。

例1-9　$AB + \overline{A}C + BC = AB + \overline{A}C$　　同样有对偶式

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

要证明两式相等，也可通过证明它的对偶式相等来完成，有时后者的证明更加容易。

以上介绍了5种运算符号，为了不致引起混乱，特作如下约定：

(1)　“·”的结合能力最强。

(2)　“⊕”和“⊙”的结合能力次之。⊕和⊙的结合能力一样，运算时彼此不分先后。

(3)　“+”的结合能力最弱。

不使用括号时，一律按结合能力的强弱次序进行运算，强者优先而弱者在后。仅在要求结合能力弱的运算先进行的场合才用括号，括号的结合能力特别强。

## §1.4　逻辑函数表示法

逻辑函数常用真值表、表达式、卡诺图、逻辑图和波形图来表示。

真值表的优点是直观、容易从实际逻辑问题中抽象出来、也可由表达式算出来。缺点是繁且不能用逻辑代数进行运算。

逻辑图是用逻辑符号表示基本单元电路而连成的图。逻辑图比较接近工程实际。

工作波形图为输入、输出相对于时间关系，用高、低电平描述的图形，形象、直观。

本节着重讨论逻辑函数表达式及卡诺图。

### 1.4.1 逻辑函数表达式

函数表达式书写方便，便于用逻辑代数进行运算。缺点是不直观。

函数表达式有一般式和标准式之分。

**一、逻辑函数一般式**

逻辑函数一般式有 5 种。借助于摩根定理和分配律，可以实现它们之间的相互转换。例

$$Y = AB + \overline{B}C \qquad \text{与或表达式}$$

$$= (A+\overline{B})(B+C) \qquad \text{或与表达式}$$

$$= \overline{\overline{AB}\,\overline{\overline{B}C}} \qquad \text{与非与非表达式}$$

$$= \overline{\overline{A+\overline{B}}+\overline{B+C}} \qquad \text{或非或非表达式}$$

$$= \overline{\overline{A}B+\overline{B}\,\overline{C}} \qquad \text{与或非表达式}$$

与或表达式和或与表达式是逻辑函数的两种基本形式，理论分析经常使用。尤其是与或表达式物理意义明确，书写方便，用得更加普遍。

与非表达式和或非表达式都能用同一种电路实现其逻辑功能，工程上普遍采用。

与或非表达式能够实现单级逻辑，在复杂系统中应尽可能多用它，这样可以节省器材。

**二、逻辑函数标准式**

逻辑函数的一般式是任意的，而逻辑函数的标准式则是唯一的。它们是标准与或式和标准或与式。

1． 标准与或式　任何逻辑函数利用互补律和分配律都可表示成标准与或式。例

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

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

在这个函数 $Y(A,B,C)$ 的与或式中，每一个乘积项均包含与此组变量相同个数的因子，且每个因子不是以原变量就是以反变量在乘积项中出现一次。这种包含所有变量的乘积项叫做最小项。对于三变量，有$2^3$种取值组合，故有八个最小项，它们是

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

一般，$n$ 变量有 $2^n$ 个最小项。由最小项构成的和式叫做逻辑函数的标准与或式或最小项表达式。

为了叙述和书写方便，对最小项进行编号。其方法是，把编号与变量的取值组合对应起来，以便容易从编号想到它的名称。为此，原变量取值为 1，反变量取值为 0。于是，变量取值组合的 2 进制值，就是最小项的代号。例 $A\overline{B}C=(101)_2=(5)_{10}$，记作 $m_5$ 或 5。对于本例逻辑函数的最小项表达式为

$$Y(A,B,C) = \sum m(1,4,5,6,7)$$