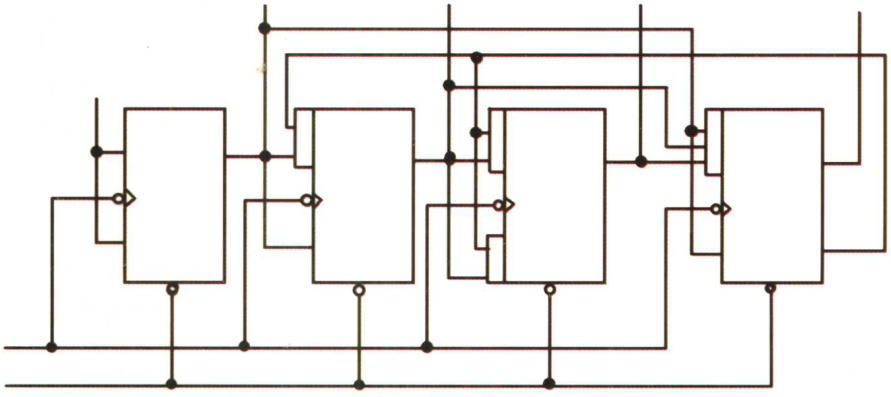
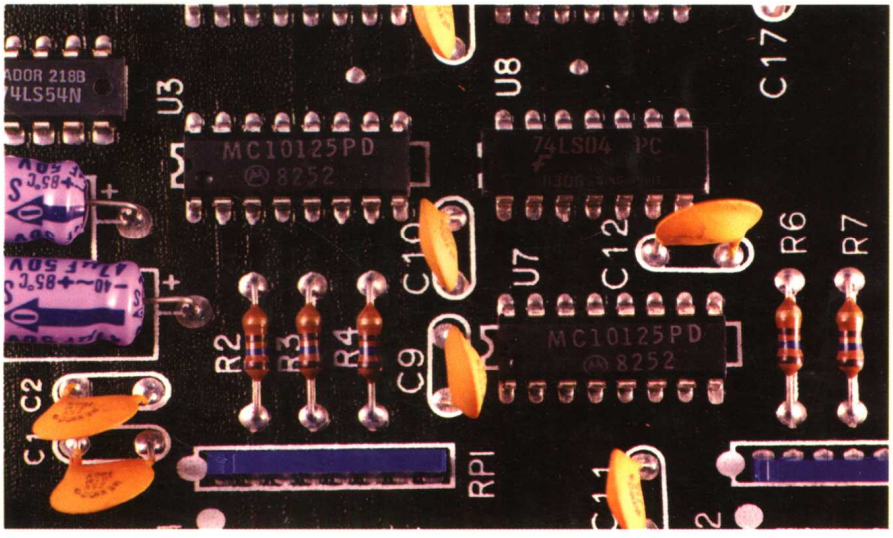
职业技术教育教材

电子技术基础（下册）

赵永良　主编

DIANZI JISHU JICHU





中国轻工业出版社

职业技术教育教材

电子技术基础

下　册

赵永良　主编
梁怀璧　主审

中国轻工业出版社

图书在版编目（CIP）数据

电子技术基础．下册/赵永良主编．—北京：中国轻工业出版社，2001.7
职业技术教育教材
ISBN 7-5019-3214-X

Ⅰ．电… Ⅱ．赵… Ⅲ．电子技术-技术教育-教材 Ⅳ．TN

中国版本图书馆 CIP 数据核字（2001）第 23663 号

责任编辑：孟寿萱　责任终审：滕炎福　封面设计：赵小云
版式设计：赵益东　责任校对：燕　杰　责任监印：崔　科

*

出版发行：中国轻工业出版社（北京东长安街 6 号，邮编：100740）
网　　址：http：//www.chlip.com.cn
联系电话：010—65241695
印　　刷：中国人民警官大学印刷厂
经　　销：各地新华书店
版　　次：2001 年 7 月第 1 版　2001 年 7 月第 1 次印刷
开　　本：787×1092　1/16　印张：7.5
字　　数：173 千字　印数：1—5000
书　　号：ISBN 7-5019-3214-X/TN・009
定　　价：13.00 元

前　言

为适应机电技术应用专业的专业建设和教材改革的需要，由原国家轻工业局机电技术应用专业建设指导委员会和全国轻工中专机电技术应用学会组织编写该专业的系列教材，《电子技术基础》是其中之一。

《电子技术基础》分上、下两册。本书为下册，内容为数字电子技术。作为专业系列教材，本书的内容既要满足本专业对电子技术知识的要求，又要能够与本系列的其他教材相衔接。同时，由于电子技术的教材已有很多，所以作者在编写本书时力求能体现针对性、实用性和先进性。编写的基本思路是：数制和转换—基本功能电路—集成电路应用。主要内容包括：数制及转换、基本门电路、组合逻辑电路、触发器及谐振电路、大规模集成电路、A/D 和 D/A 转换以及数字电子技术应用等。在内容编排上，既考虑到系统性，又注意到对本专业的适用性，注重介绍电子技术在机电一体化设备和产品中的应用实例，并希望能体现电子技术发展快的特点，因此引入了一些较新的内容（如中大规模集成电路及其应用等），为学生进一步学习微机系统的知识打下较好的基础。在叙述方法上，按照当前中等职业教育教学改革的要求，注意突出基本概念、基本原理和基本分析方法，以介绍电路的集成器件的外部特性及应用知识为主，并适当降低理论深度，降低对设计、计算能力的要求，相应加强了对分析、应用和电子电路读图能力的要求，注意加强对学生运用知识的能力、继续学习能力和创新精神的培养。

本教材适合当前对中等职业技术人才的培养要求，使学生获得高素质劳动者和中级专门人才必须具有的电子技术基本理论、基本知识和基本技能，并为后续课程的学习准备必要的知识，为今后从事实际工作打下必要的基础。全书的理论教学时数为 150～180，下册的教学时数为 60～80，其中基础内容为 60 学时，拓宽内容为 10 学时，选用内容为 10 学时，可供不同学制、专业选择使用。

本书由北京轻工职业技术学院赵永良主编。参加本书编写工作的有赵晓文、肖永生、沈伟东、李建宏、黄蓉、李惠芳。北京航空航天大学梁怀璧教授担任本书的主审，他对本书的结构和内容提出了许多宝贵意见，特表示感谢。

由于时间紧迫，编者水平有限，本书的错漏、不当之处在所难免，诚挚地欢迎广大教师和读者提出宝贵意见。

编者

2001 年 1 月

目　录

第一章　数字电路基础知识

本章介绍十进制数、二进制数、十六进制数及BCD码，以及它们之间的相互转换，并介绍三种基本逻辑运算和逻辑函数的定理以及逻辑函数的化简方法：代数化简法和图形化简法。

第一节　数制码制及其转换

一、数　　制

（一）十进制（Decimal system）

在日常生活中，人们习惯于采用十进制数，它是由0、1、2、3…9这十个数码按照一定的规律排列起来表示数值的大小，在计数时采用“逢十进一”的规则。

例如，6931这个四位数可以表示为

$$[6931]\ \mathrm{D}=6\times10^3+9\times10^2+3\times10^1+1\times10^0$$

在计算机系统中，[6931] D又常写作6931D。

通常把10^3、10^2、10^1、10^0称为对应位的权或称为位权。

一般地，任意一个十进制的正整数可以表示为

$$N_{\mathrm{D}}=K_{n-1}\times10^{n-1}+K_{n-2}\times10^{n-2}+\cdots+K_0\times10^0$$

$$=\sum_{i=0}^{n-1}K_i\times10^i$$

式中，K_i代表第i位的系数，可取0，1，2…9这十个数码，而10^i为第i位的权，除了以10为基数的十进制外，还有取其他基数的计数制，如十六进制、八进制、二进制等。

（二）二进制（Binary system）

在数字电路或计算机中，数字是用电路中的引线端上的电压（在数字电路中称为电平）的两种状态（高或低）来表示的，所以，数的处理与运算必须按二进制的计数体制来进行。二进制只有0和1两个数码，采用“逢二进一”的规则。

例如，1011这个四位二进制数可以表示为

$$[1011]\ \mathrm{B}=1\times2^3+0\times2^2+1\times2^1+1\times2^0$$

在计算机系统中，[1011] B又常写作1011B。

同样，任意一个二进制正整数可以表示为

$$N_{\mathrm{B}}=K_{n-1}\times2^{n-1}+K_{n-2}\times2^{n-2}+\cdots+K_0\times2^0$$

$$=\sum_{i=0}^{n-1}K_i\times2^i$$

式中N_{B}表示二进制数，K_i表示第i位的系数，只取0和1中的任一个数码，2^i为第

i 位的权。

在数字电路中，一般用高电平来表示数字1，用低电平来表示 0。

（三）十六进制（Hexadecimal system）

十六进制中，应该有 16 个数字符号，我们借用 0、1、2、3、4、5、6、7、8 这 9 个阿拉伯数字，再借用 6 个拉丁字母 A、B、C、D、E、F，即可凑足所需的数字符号了。这里的 A 代表十六进制数，相当于十进制的 10。

例如，10AF 这个四位十六进制数可以表示为

$$[10AF]\ H=1\times16^3+0\times16^2+10\times16^1+15\times16^0$$

在计算机系统中，[10AF]H 又常写作 10AFH。如果一个十六进制数的最高位是 A，B，C，D，E，F 中的一个，如 [AF] H，则按照这种写法书写时，还应在最高位前加一个 0，成为 0AFH。

同样，任意一个十六进制正整数可以表示为

$$N_H=K_{n-1}\times16^{n-1}+K_{n-2}\times16^{n-2}+\cdots+K_0\times16^0$$

$$=\sum_{i=0}^{n-1}K_i\times16^i$$

式中 N_H 表示十六进制数，K_i 表示第 i 位的系数，可取 0～F 中的任一个数码，16^i 为第 i 位的权。

二、码　　制

二进制数码不但能用来表示一定的数量，还可以用来表示各种文字和符号信息，通常称这种二进制数码为代码（Code）。建立这种代码与文字、符号或特定对象之间一一对应关系的过程就称为编码（Coding）。

数字电路中常用到的码制，是所谓的二一十进制编码，常简称为 BCD（Binary Coded Decimal）码，指的是用 4 位二进制数来表示一位十进制数的编码方式。

由于四位二进制数码可以表示 16 种不同的组合状态，所以在表示 1 位十进制数码（十个数码）时，只需选择其中的 10 种状态的组合，其余 6 种是无效的。通常按选择的方式不同，可以得到不同的二一十进制编码，如 8421BCD 码，5421BCD 码，余 3 码和格雷码等。一般可将其分为有权码和无权码两大类，例如 8421 码是一种有权码，8421 就是指这种编码中各位的权分别是 8、4、2、1。此外，5421 码也属有权码，而余 3 码和格雷码则是无权码。对于有权码来说，由于各位均有固定的权，因此二进制数码所表示的十进制数就容易识别。在数字电路及计算机电路中，最常用的是 8421BCD 码。以下为一个用 8421BCD 码来表示多位十进制数的例子：

$$[450]\ D=[0100\ 0101\ 0000]\ BCD$$

三、数制及码制的转换

（一）二进制数转换成十进制数

二进制数转换成十进制数并不困难，只要将其按权展开就可以了。

例如，将二进制数 1010 转换成十进制数：

$$[1010]\ B = 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0$$
$$= [10]\ D$$

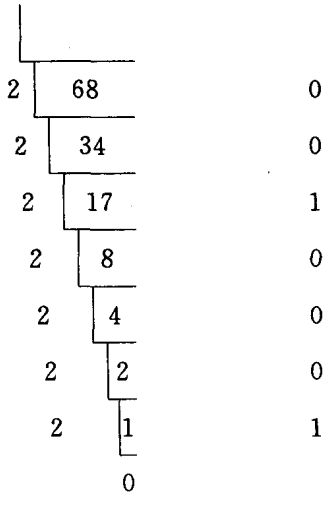
（二）十六进制数转换成十进制数

方法同上，只要将其按权展开就可以了。

（三）十进制数转换为二进制数

要将十进制数转换成二进制数，可以采用除 2 取余法，即用 2 不断地去除十进制数，直到最后商为 0 为止，将所得到的余数以最后一个余数为最高位，第一个余数为最低位，依次排列便得到相应的二进制数。

例如，将 68 转换为二进制数：



$$[68]\ D = [1000100]\ B$$

（四）十六进制数与二进制数之间相互转换

十六进制数的每一位的取值有 0～F 共 16 种可能，故其 1 位相当于二进制的 4 位（$2^4=16$）。所以，当由十六进制向二进制转换时，只需将其每 1 位换算成二进制的 4 位即可；而当二进制向十六进制转换时，则须从二进制的最低位开始，每 4 位为一组，换算成十六进制的 1 位。

例如，将二进制数 11110000 转换成十六进制数：

$$11110000B = 0F0H$$

将十六进制数 3AH 转换成二进制数：

$$3AH = 00111010B = 111010B$$（高 2 位的 0 舍去）

（五）十进制数与 8421BCD 码之间相互转换

十进制数与 8421BCD 码之间的相互转换与十六进制数与二进制数之间相互转换很相似，所不同的是当十进制数向 BCD 码转换时，产生的每一组的二进制数都必须是 4 位，高位的 0 不能舍去；另外，每一组之间应有空格隔开。

例如，将十进制数 139D 转换成 8421BCD 码：

$$139D = [0001\ 0011\ 1001]\ BCD$$

第二节　逻辑代数

所谓逻辑是指“条件”与“结果”的关系，这种关系又称为逻辑关系，“条件”和“结果”都称为逻辑变量，其中“条件”为自变量（常简称变量），“结果”为应变量（常称为函数）。逻辑变量的值称为真值，其取值只有两种可能，一种为“真”（True）（或称“成立”），另一种为“假”（False）（或称“不成立”）。一般用“1”表示“真”，“0”表示“假”，显然，这里的“1”和“0”都已失去了原来的数字的含义。

逻辑代数就是数学中用来描述逻辑关系、反应逻辑变量运算规律的一个分支。下面介绍三种基本逻辑运算、逻辑代数的定律和规则以及逻辑函数的化简。

数字电路不仅能够对二进制数进行处理，而且由于其电平信号的二值性，它也可以对逻辑关系进行描述。所以，数字电路又可称为逻辑电路，在逻辑电路中，逻辑变量是电路中各引线端的电平，若为高电平，则记作 1；若为低电平，则记作 0。自变量指的是逻辑电路的输入端的电平，函数则指的是输出端的电平。

一、基本逻辑运算

基本逻辑运算有“与”（AND）、“或”（OR）、“非”（NOT）三种，它们可以由相应的逻辑电路来实现。

（一）与运算（逻辑乘）

与运算用来表示这样一种逻辑关系：只有决定某一结果的全部条件都满足时，这个结果才会发生。下面举例说明这种逻辑关系。

现象：某保险柜需两把钥匙同时插入才能打开，而它们由 A，B 二人分别保管。逻辑关系：结果“保险柜能打开否”（记作函数 Y）与条件“A 在场否”（记作变量 A）及条件“B 在场否”（记作变量 B）之间存在着这样的逻辑关系：只有当 A 和 B 都成立时，Y 才成立。这种逻辑关系可用以下逻辑函数式来表示：

$$Y=A\times B \quad 或\ Y=AB$$

其中 A、B 之间的运算称为“与运算”或“逻辑乘”。

我们可以将 A、B 的真值的所有组合及对应的函数 Y 的真值一一排列出来，从而形成一个表格，这个表格称为与运算的“真值表”，见表 1-1。

显然有如下关系成立：

$$0\times 0=0$$
$$0\times 1=0$$
$$1\times 0=0$$
$$1\times 1=1$$
$$A\times 0=0$$
$$A\times 1=A$$
$$1\times A=A$$

也可以有多个变量相与，如 $Y=A\times B\times C\times D$

表 1-1　与运算真值表

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

能够实现与运算的逻辑电路是与门（AND Gate）电路，二输入端的与门的逻辑符号如图 1-1 所示。显然，与门的逻辑关系应该是：只有所有的输入端都加上高电平信号 1 时，输出才为高电平 1，而只要有一个输入为 0，则输出必为 0。

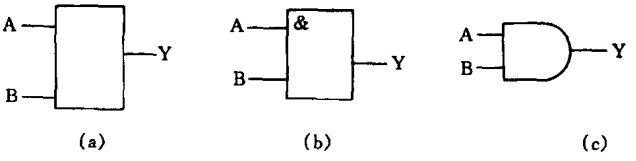


图 1-1　与门的逻辑符号

(a) 国标符号

(b) 国内常用符号

(c) 国外常用符号

（二）或运算（逻辑加）

或运算用来表示这样一种逻辑关系：决定某一结果的各个条件中，要有一个条件满足，或者几个条件满足，这个结果就会发生。

下面以一个白炽灯的控制电路（注意，此电路并不是本书中所研究的逻辑电路）来说明或运算。

图 1-2 中两个并联开关控制一个灯泡，当 A 和 B 两个开关中有一个接通，灯泡 EL 便亮，若将开关接通记作“1”，开关断开记作“0”，灯亮记作“1”，灯灭记作“0”，此逻辑关系的真值表见表 1-2。

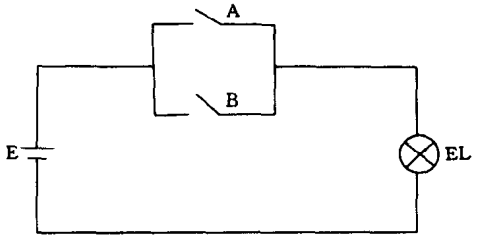


图 1-2　白炽灯控制电路

表 1-2　或运算真值表

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

或运算可以用下式表示：

$$Y=A+B$$

显然，如下的逻辑关系是成立的：

$$0+0=0$$
$$0+1=1$$
$$1+0=1$$
$$1+1=1$$
$$A+A=A$$
$$A+0=A$$
$$A+1=1$$

也可以有多个变量相或，如 $Y=A+B+C+D$。

实现逻辑加的电路是或门（OR Gate），或门的逻辑符号如图 1-3，对于或门来说，只要有一个输入为 1，则输出为 1，只有所有的输入皆为 0，输出才为 0。

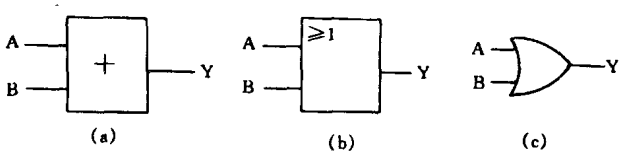


图 1-3　或门的逻辑符号

（三）非运算（逻辑反）

非运算表示一事物相反的情况出现时为“真”。如图 1-4，开关 A 接通，灯反而不亮。

非运算真值表如表 1-3 所示，并可用如下逻辑表达式表示：

$$Y=\overline{A}$$

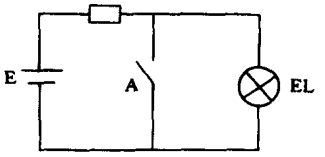


图 1-4　白炽灯控制电路之二

表 1-3　非运算真值表

A	Y
0	1
1	0

通常，A 称为“原变量”，$\overline{A}$称为“反变量”。显然，有如下关系成立：

$$\overline{0}=1$$

$$\overline{1}=0$$

$$\overline{A}+A=1$$

$$\overline{A}\times A=0$$

$$\overline{\overline{A}}=A$$

实现逻辑非的电路是非门（NOT Gate），有时也称反相器（Inverter）。非门的逻辑符号如图 1-5 所示，非门的输入和输出永远是相反的。

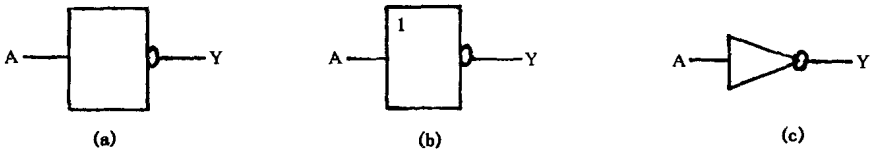


图 1-5　非门的逻辑符号

二、逻 辑 函 数

将逻辑变量 A，B，C，…用有限个与、或、非逻辑符号按某种逻辑关系连接起来，所得到的表达式 $Y=f$（A、B、C、…）称为逻辑函数。

下面列举几种常见的逻辑函数及能完成其逻辑功能的逻辑门。

（一）与非函数（与非门）

与非（NAND）门是一种常见的逻辑门，其逻辑符号如图 1-6，与非门可看成一个与门后接一个非门。

与非门的逻辑函数式为

$$Y=\overline{AB}$$

其真值表如表 1-4 所示。

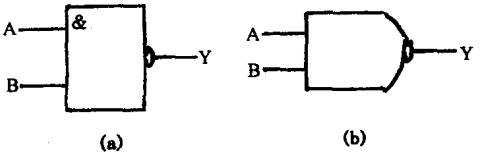


图 1-6　与非门的逻辑符号

表 1-4　与非运算真值表

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

（二）或非函数（或非门）

或非（NOR）门逻辑符号如图 1-7，它可以看成是一个或门后接一个非门，或非门真值表如表 1-5，其逻辑函数式为

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

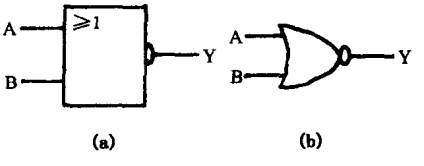


图 1-7　或非门的逻辑符号

表 1-5　　或非门真值表

A	B	Y	A	B	Y
0	0	1	1	0	0
0	1	0	1	1	0

（三）同或函数（同或门）

同或（X-NOR）门是将两路输入信号 A 和 B 进行比较，判断它们是否相同，当 A 和 B 相同时，输出 Y 为 1，当 A 和 B 不同时，输出 Y 为 0，它的真值表如表 1-6。

表 1-6　　同或门真值表

A	0	1	0	1
B	0	0	1	1
Y	1	0	0	1

同或门对应的函数式为

$$Y=\overline{A}\times\overline{B}+A\times B$$

或写成　　$Y=A\odot B$

（四）异或函数（异或门）

异或（X-OR，E-OR）门是将两路输入信号进行比较，判断它们是否不同，当两路输入信号不同时，输出为 1；反之，则为 0。

异或门的逻辑函数式为

$$Y=A\times\overline{B}+\overline{A}\times B \quad 或 \quad Y=A\oplus B$$

三、逻辑代数的基本定律

以上介绍了与、或、非三种逻辑运算及常用逻辑函数，下面介绍逻辑代数的基本定律常用规则及常用公式。

（一）基本定律

1. 交换律　　$A+B=B+A$

2. 结合律　　$A+(B+C)=(A+B)+C$

3. 分配律　　$A(B+C)=AB+AC$

$A+B\times C=(A+B)\times(A+C)$

4. 吸收律　　$A+AB=A$

5. 0－1 律　　$A+1=1$

$A+0=A$

$A\times 1=A$

$A\times 0=0$

6. 互补律　　$A+\overline{A}=1$

$\overline{A}\times A=0$

这些基本定律都可以由真值表得到证明。

（二）常用规则

1. 代入规则

任何一个含有变量 A 的等式，如果将所有出现 A 的位置都代之以一个逻辑函数 F，则等式依然成立。此规则称为代入规则。

例如：　　$A+B=B+A$

若　　$B=CD$

则　　$A+CD=CD+A$

2. 反演规则（摩根定理）

反演规则是逻辑代数的一个重要规则，其公式如下：

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

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

该公式也可以由真值表得到证明。

（三）常用公式

利用逻辑代数的基本定律和规则，可以得到一些常用的公式。

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

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

$=A$

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

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

$=A+B$

3. $\overline{A\overline{B}+\overline{A}B}=AB+\overline{A}\,\overline{B}$

证明：$\overline{A\,\overline{B}+\overline{A}B}=\overline{A\,\overline{B}}\times\overline{\overline{A}B}$

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

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

四、逻辑函数的化简

通常直接根据实际逻辑问题归纳出来的逻辑关系及其对应的逻辑函数往往不是最简，因此，有必要对其进行化简。

例如，某函数 $Y=A\,\overline{B}C+A\,\overline{B}\,\overline{C}$，经过化简后变为 $Y=A\,\overline{B}$，大大地减少了所需要的逻辑电路。

下面介绍两种化简逻辑函数的方法：代数化简法和图形化简法。

（一）代数化简法

代数化简法也称公式化简法，它是利用逻辑代数的基本公式和常用公式对逻辑函数进行化简。常用的方法如下：

1. 并项法

利用公式 $AB+A\,\overline{B}=A$ 消去多余项。

〔例 1-1〕化简函数 $Y=A\,\overline{B}C+A\,\overline{B}\,\overline{C}$。

解：

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

2. 吸收法

利用公式 $A+AB=A$ 去掉多余项。

〔例 1-2〕化简函数 $Y=AB+AB\,(C+D)$。

解：$$Y=AB\,(1+C+D)=AB$$

3. 消去法

利用公式 $A+\overline{A}B=A+B$ 消去多余项。

〔例 1-3〕化简函数 $Y=AB+B\,\overline{C}+A\,\overline{C}$。

解：

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

4. 配项法

利用公式 $A=A\,(B+\overline{B})$ 将表达式中不能利用公式化简的某些积项变成两项，然后再利用公式化简。

〔例 1-4〕化简函数 $Y=A\,\overline{B}+B\,\overline{C}+\overline{B}C+\overline{A}B$。

解：

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