高等学校教学参考书

逻辑电路的分析与设计

杨福生 编

人民教育出版社

本书简明地介绍了怎样分析和设计逻辑电路。它以布尔代数和卡诺图作为主要工具，除阐明基本概念外，比较系统地介绍了一些实用的分析和设计的步骤、方法。全书共分五章。在简要地叙述逻辑电路的基本概念、逻辑代数的运算方法以后，讨论了组合电路和顺序电路。顺序电路按计数电路、同步电路及异步电路的次序介绍。

本书编者系清华大学电机系应用电子学及电工学教研室教师。本书是编者根据本人给青年教师讲课的讲稿整理而成的。内容深入浅出，例题较多，易学实用。

本书对稍有一些数字电路基础知识的读者进一步学习逻辑电路的分析与设计是适宜的。可供工科大专院校非电类专业学生在深入学习电工学时参考，可供电工学教师参考，也可供电类专业学生及工程技术人员参考。

本书责任编辑　王缉惠

高等学校教学参考书
逻辑电路的分析与设计
杨福生 编
*
人民教育出版社出版
新华书店北京发行所发行
人民教育出版社印刷厂印装
*
开本 787×1092 1/16　印张 8.75　字数 188,000
1981 年 10 月第 1 版　1982 年 11 月第 1 次印刷
印数 00,001—13,000
书号 15012·0367　定价 0.84 元

目　录

第一章　基本概念……1
1.1　概述……1
1.2　逻辑变量和逻辑代数……2
（一）逻辑代数……2
（二）基本逻辑算子……3
1.3　逻辑代数的基本运算规则和定理……5
（一）基本运算规则……5
（二）基本定理……6
（三）对偶原理……7
（四）反演定理(摩根定理)……7
（五）香农定理……8
1.4　逻辑函数的表示方法……9
（一）真值表……9
（二）图象表示……10
（三）逻辑代数表达式……11
1.5　逻辑表达式的化简……13
（一）利用逻辑代数基本公式化简……13
（二）用卡诺图化简……14
（三）约束条件的利用……16
（四）本节总结……18
1.6　列表化简法……21
习题……23
第二章　组合电路的分析与设计……25
2.1　概述……25
2.2　组合电路的分析……25
（一）逐级推导电平法……25
（二）逻辑函数推导法……25
（三）分别分析在控制输入不同组合下输入与输出的关系……26
2.3　组合电路设计的一般步骤及应用举例……27
（一）码的互换……27
（二）造表……28
（三）数字运算电路……30
2.4　只读存贮器的编程……33
2.5　多输出的组合电路……38

2.6 禁止门及封锁概念的应用 …… 42
(一) 基本概念 …… 42
(二) 应用封锁概念分析多级与非门电路 …… 44
(三) 应用封锁概念设计多级与非门电路 …… 45
2.7 异或门的应用 …… 50
(一) 有关异或运算的一些基本公式 …… 50
(二) 利用异或门构成组合电路 …… 51
附录 数字量的二-十进制编码 …… 53
习题 …… 55
第三章 计数电路的分析与设计 …… 58
3.1 概述 …… 58
3.2 二进制计数器 …… 60
3.3 同步计数电路的分析 …… 63
3.4 同步计数电路的设计 …… 64
3.5 移位寄存器型计数电路 …… 67
3.6 异步计数电路的设计 …… 71
(一) 反馈置零法 …… 71
(二) 脉冲反馈法 …… 72
(三) 阻塞导引法 …… 72
(四) 利用因子分解 …… 77
习题 …… 79
第四章 同步顺序电路的分析与设计 …… 80
4.1 概述 …… 80
(一) 同步顺序电路的基本组成 …… 80
(二) 同步顺序电路逻辑功能的描述方法 …… 81
4.2 同步顺序电路的分析 …… 83
4.3 状态表的电路实现 …… 85
(一) 实现举例 …… 86
(二) 设计步骤总结 …… 88
(三) 编码问题 …… 88
4.4 状态表的拟制与化简 …… 90
(一) 设计举例 …… 90
(二) 状态的合并 …… 93
4.5 设计举例 …… 95
习题 …… 101
第五章 异步顺序电路的分析与设计 …… 103
5.1 概述 …… 103
5.2 异步顺序电路的分析 …… 104
(一) 分析步骤 …… 104

(二) 讨论 ……110
(i) 竞争问题 ……110
(ii) 循环现象 ……112
(iii) 冒险现象 ……112
5.3 异步顺序电路的设计 ……113
(一) 原始状态表或状态流程表的拟制 ……113
(二) 状态的化简 ……115
(三) 状态编码 ……116
(四) 组合电路的实现 ……118
(五) 构作输出电路 ……120
(六) 设计步骤总结 ……121
5.4 设计举例 ……122
附录 冒险现象 ……127
习题 ……128

此为试读,需要完整PDF请访问: www.ertongbook.com

第一章 基本概念

1.1 概 述

逻辑电路也叫开关电路或数字电路。它区别于模拟电路的特点是：电路中的变量(一般指电压)只能取两个互异的状态，而且变量的变化和电路的动作只发生于某些离散的时刻。逻辑电路由诸如门电路和触发器之类的基本逻辑部件组成。本书不讨论逻辑部件的原理和设计，并且假定读者已经掌握了这方面的基本知识(这方面内容可以参考脉冲数字电路的教科书)。本书任务是介绍如何分析和设计用基本逻辑部件构成的逻辑电路。

逻辑电路可以大致分成两类：

1) **组合电路** 也叫无记忆的开关电路。它的特点是：输出的现时值由输入的现时值唯一地决定。象加法器、译码器、数据选通器、数字比较器等都属于这一类。组合电路的基本组成部件是各种逻辑门，如：与门、或门、非门、与非门、或非门、异或门等。

2) **顺序电路** 这种电路的特点是：输出不只决定于输入的现时值，而且还和输入过去的历史以及电路内部的初始状态有关。因此，同样的输入可以产生不同的输出。计数器是这类电路的典型例子：当计数器进行计数时，它的输入每次都一样，都是一个计数脉冲；可是输出(指计数器的计数结果)却每次不同。顺序电路的方框结构可以用图1.1表示。它由组合电路和记忆电路两部分组成。记忆部分用来存贮电路过去的有关信息。电路的现时输出不只决定于输入的现时值，而且还和被记忆下来的信息有关。记忆部分的基本部件是触发器、磁芯、继电器等二态器件(严格说，触发器也是由逻辑门加反馈构成的，但是我们分析设计时将以触发器作为记忆部分的基本部件。因为成块触发器组件已大量生产)。

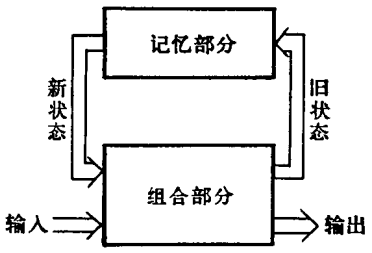


图 1.1

顺序电路又可以根据信号变化的特点分成同步和异步两类。同步电路中电路状态变化的时刻由整步脉冲(也叫时钟脉冲)来控制，以保证动作同步。异步电路则无此特点，因此电路各处信号变化时刻往往参差不齐。异步电路的结构一般比同步电路简单；但是，由于动作不同步就容易产生一些特殊问题(如：竞争、冒险等)，使它的分析和设计更复杂一些。

本书任务是介绍分析和设计这两类电路的基本概念和方法。在内容取舍上有两个主要考虑：

1) 不是讲具体的数字控制和数字测量技术。因此不对具体仪器作详细介绍，而是着重于讲解分析方法。

2) 方法的介绍又着重于实用性，希望读者能学会一些简单实用的方法，以便具有处理一些

1110531

不太复杂的实际问题的能力。因此不追求数学证明的严谨，而着重于基本概念的阐述。为了使读者学后具有进一步自学的能力，一些基本名词和定义仍要交代清楚。

1.2 逻辑变量和逻辑代数

（一）逻辑代数 逻辑代数又称布尔代数，本来是人们利用数学方法研究思维规律发展起来的。它是上世纪中叶爱尔兰数学家 George Boole 创始的。随着电子技术，特别是数字技术和逻辑机器的发展，人们发现布尔代数的运算规则可以应用于这种电路的分析与设计。一方面，它往往能使我们将逻辑要求用简洁的数学形式表达出来；另一方面，它又能帮助我们将一些简单逻辑电路设计出来。这样就大大扩展了这一数学方法的应用领域。目前，它在自动化技术及电子计算机逻辑设计中都有广泛应用。

在学习脉冲数字电路时读者已经有了布尔代数的初步概念。如所已知，它是一种适用于逻辑推理的代数方法：

以符号X代表任一事件或命题，并以$X=1$代表此事件发生或此命题成立，$X=0$代表此事件不发生或此命题不成立。另外，用“与”“或”“非”三个逻辑概念来表示事件或命题之间的联系：

1. 与（也叫逻辑乘）：当命题X_1与命题X_2同时成立，命题Z才成立，便称Z是X_1和X_2的与函数，表示成

$$Z=X_1\cdot X_2\text{（或 }Z=X_1\times X_2\text{，或}Z=X_1\wedge X_2\text{）} \tag{1-1}$$

2. 或（也叫逻辑加）：当命题X_1或X_2任意有一个成立或同时成立，命题Z就成立，便称Z是X_1和X_2的或函数，表示成：

$$Z=X_1+X_2 \quad \text{（或 } Z=X_1\vee X_2\text{）} \tag{1-2}$$

3. 非：当命题X_1不成立，命题Z就成立；反之，前者成立后者就不成立，便称Z是X_1的非函数（或叫反函数），表示成：

$$Z=\bar{X}_1 \quad \text{（或 } Z=\dot{X}_1, Z=X_1'\text{）} \tag{1-3}$$

生产和生活中的各种关系常常可以用逻辑代数来表示。例如：只当公共汽车的前门与后门都关上后汽车才能开行。如果用X_1代表前门关闭这一事件，X_2代表后门关闭这一事件，Z_1代表汽车开行这一事件，则上述关系可以表示成：

$$Z_1=X_1\cdot X_2 \tag{1-4}$$

又，前后门里有任何一个打开，乘客便可以上车。如果用Z_2代表乘客上车这一事件，则这个关系可以表示成：

$$Z_2=\bar{X}_1+\bar{X}_2 \tag{1-5}$$

这些表示逻辑关系的表达式称为逻辑函数。式中各字母叫逻辑变量。逻辑变量的可能取值只有两个。为了便于和常用代数作对比，常用“0”和“1”代表这两个取值。注意，它只是一种符号表示，并无定量含义。从原理上看，变量取值的具体数值并不重要，重要的是它究竟处于两个状态的哪一组中。有时，为了避免和普通代数符号混淆，用T(True)和F(False)来代表这两个状态。有时，为了联系电位情况，又用H(High)和L(Low)来代表。如果变量$=1$定义为事件发生，

变量=0 定义为事件不发生，并且按下述规则进行逻辑代数的运算：

$$0+0=0$$

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

$$0\cdot 1=1\cdot 0=0\cdot 0=0$$

$$1\cdot 1=1$$

$$\bar{1}=0 \quad , \quad \bar{0}=1$$

那末上述事件间关系就被逻辑函数代数地表现了出来。具体地说：

(1-4)式指出：只当 $X_1=1$ 与 $X_2=1$ 同时成立才有 $Z_1=1$。在输入的其他组合下都有 $Z=0$。这便意味着：只当前后车门都关闭时汽车才能开行。

(1-5)式指出：当 $X_1=0$(因此 $\bar{X}_1=1$)或 $X_2=0$(因此 $\bar{X}_2=1$)时就有 $Z_2=1$。这意味着：前后车门中任何一个打开乘客便可上车。

逻辑函数还可以用来表示开关电路的控制关系，见图 1.2。图上用 $X_i=1$ 代表触点 X_i 动作，用 $Z=1$ 代表负载通电。

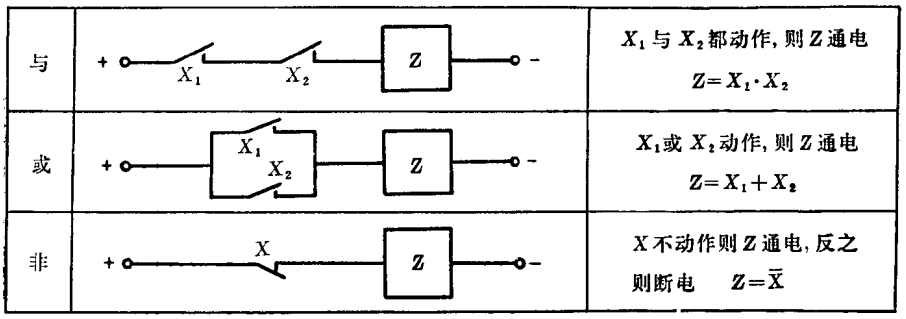


图 1.2

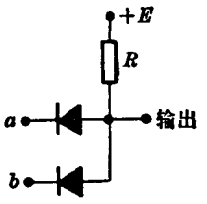


图 1.3

在电子开关电路中，逻辑关系是以各种逻辑门电位的高低来实现的。如果以高电平为“1”，低电平为“0”，便称为正逻辑系统；反之便称为负逻辑系统。读者已经知道：一个具体的门电路，实现的究竟是何种逻辑功能与采用的是正逻辑系统还是负逻辑系统有关。例如，图 1.3 对正逻辑系统而言是与门(a、b 两端同时等于“1”即高电平，输出才等于“1”)，可是它对负逻辑系统而言却是或门(a、b 两端任何一个等于“1”即低电平，输出便等于“1”。)

n 个逻辑变量结合起来构成一组逻辑变量的集合。其中每个变量取值都可以是 0 或是 1。变量取值的每一种组合称为一个“状态”。显然，n 个变量构成的集合可以有 2^n 个不同状态。例如，当 $n=2$ 时，$2^n=4$。即：两个逻辑变量可以组合成四种状态，它们是：$X_1X_2=00, 01, 10, 11$。

(二) 基本逻辑算子 根据实际条件，各逻辑变量间可能有各种联系，可以用逻辑函数 $Z=f(X_1, X_2\cdots X_n)$表示；其中 X_1、$X_2\cdots X_n$ 是输入变量，Z 是输出变量。可以证明，复杂的逻辑函数总可以用若干个最基本的逻辑算子组成。

最基本的逻辑算子如下：

1. 单变量情况：有四个基本算子，如下(表 1.1)：

表 1.1

输入	输出
X	Z_0
0	0
1	0

称为"零函数"，特点是：不论输入是什么，输出恒为0。

输入	输出
X	Z_1
0	0
1	1

称为"同函数"，特点是：输出和输入总是相同的。

输入	输出
X	Z_2
0	1
1	0

称为"非函数"，特点是：输出总是和输入相反的。

输入	输出
X	Z_3
0	1
1	1

称为"幺函数"，特点是：不论输入是什么，输出恒为1。

2. 二变量情况：两个变量可以组合成四种状态。对应于这四种输入状态，输出可能有十六种不同响应，如下(表 1.2)：

表 1.2

输入	X_2	0	0	1	1	名称与说明	符　号
	X_1	0	1	0	1		
输出	Z_0	0	0	0	0	零函数	$Z_0=0$
	Z_1	1	0	0	0	或非	$Z_1=\overline{X_2+X_1}$
	Z_2	0	1	0	0	X_1禁止X_2	$Z_2=\overline{X}_2X_1$
	Z_3	1	1	0	0	非X_2	$Z_3=\overline{X}_2$
	Z_4	0	0	1	0	X_2禁止X_1	$Z_4=X_2\overline{X}_1$
	Z_5	1	0	1	0	非X_1	$Z_5=\overline{X}_1$
	Z_6	0	1	1	0	异或	$Z_6=X_1\oplus X_2$
	Z_7	1	1	1	0	与非	$Z_7=\overline{X_1X_2}$
	Z_8	0	0	0	1	与	$Z_8=X_1X_2$
	Z_9	1	0	0	1	同或	$Z_9=X_1\odot X_2$
	Z_{10}	0	1	0	1	同X_1	$Z_{10}=X_1$
	Z_{11}	1	1	0	1	蕴含	$Z_{11}=\overline{X}_2+X_1$
	Z_{12}	0	0	1	1	同X_2	$Z_{12}=X_2$
	Z_{13}	1	0	1	1	蕴含	$Z_{13}=X_2+\overline{X}_1$
	Z_{14}	0	1	1	1	或	$Z_{14}=X_1+X_2$
	Z_{15}	1	1	1	1	幺函数	$Z_{15}=1$

其中，有些是上面已经见过的(同、非、零、幺)，有些是大家已往熟知的(与、或、与非、或非)。需要略加解释的是：

i. 异或算子(用符号⊕表示)和同或算子(用符号⊙表示)。异或算子的特点是：当两个输入相同时输出为 0，输入不同则输出为 1。由于这一特点，技术上颇有应用，目前已生产单块异或门组件。同或算子的特点恰和异或相反：输入相同时输出为 1，否则为 0。不难看出它实际是异或函数的非函数。即：

$$\overline{A\oplus B}=A\odot B$$

ii. 禁止算子。以 Z_2(X_1 禁止 X_2)为例。由表 1.2 可见它的特点是：当 $X_1=0$ 时输出便被封锁在 0 状态下，不随 X_2 变。只有当 $X_1=1$ 时，输出才随 X_2 变，取值与 X_2 相反。

可以证明，这些基本算子并不是相互独立的，某些算子的集合可以把其他各算子全部引伸出来。例如，“与”和“非”集合、“或”和“非”集合都具有这样的特点（只要根据表 1.2 中的逻辑表达式便能证明）。具有这一特性的集合称为“完备的集合”。还可以证明，“与非”和“或非”函数本身就能构成完备集合，因此用它们作逻辑电路的基本单元是最便利的。基本算子的完备性对研制生产基本逻辑门有实际意义。应该指出：异或门本身并不构成完备的集合，必须和与门结合起来才能构成完备的集合。

任何复杂的逻辑函数总可以分解成若干基本逻辑算子的组合，因此总可以用一组完备算子的集合来实现。这是逻辑电路分析和设计的基础。用基本逻辑算子构成的逻辑函数叫逻辑表达式。如：

$$Z=\overline{X_1\cdot X_2}+\overline{X_3\cdot X_2}$$

$$Z=X_1\oplus X_2\oplus X_3+\overline{X_2X_4}$$

更复杂的例子如：

$$Z=X_1[\overline{(X_2\cdot X_3)}+\overline{(X_2+X_3)X_4}]$$

有括弧时的运算次序和普通代数的规定一样，即从最内部的括弧逐步向外运算。

1.3 逻辑代数的基本运算规则和定理

（一）基本运算规则 因为逻辑变量的取值只可能是 0 或 1，而最基本的逻辑运算可以归结成与、或、非三种。因此，基本运算规则只有以下几种：

与：$0\times1=1\times0=0$

$0\times0=0$

$1\times1=1$

或：$0+0=0$

$0+1=1+0=1$

$1+1=1$

非：$\bar{0}=1\quad\bar{1}=0$

这些基本运算规则是人为的规定。人们所以这样来规定是因为它既与开关电路或逻辑思维的推论一致，又与人们已经习惯了的普通代数运算规则相似。

由这几个基本运算规则又可以得出以下推论：

$A+0=A$　　$A+1=1$

$A\cdot0=0\cdot A=0$　　$A\times1=A$

$A+\bar{A}=1$　　$A+A=A$

$A\cdot\bar{A}=0$　　$A\cdot A=A$

$\bar{\bar{A}}=A$

这些结论可以用代入真值进行运算的方法来证明。以 $A+\bar{A}=1$ 为例：

$$\left.\begin{array}{l}\text{当 } A=1 \text{ 时，} A+\bar{A}=1+0=1\\ \text{当 } A=0 \text{ 时，} A+\bar{A}=0+1=1\end{array}\right\}\text{可见 } A+\bar{A}=1 \text{ 正确。}$$

（二）基本定理 多变量时逻辑代数的基本定理可概括如下：

1. 交换律，结合律及分配律都适用。例如：

$$A+B=B+A$$
$$A+(B+C)=(A+B)+C=(A+C)+B$$
$$A\cdot(B+C)=AB+AC$$

这些关系仍可用代入真值的方法来证明。事实上它是基本运算规则的必然推论，因为我们规定的基本运算规则本身就符合这些规律。

2. 几种形式的吸收规则。逻辑表示式区别于普通代数表达式的一个重要特点是：表达式中某些项（或某些因子）可能被其他项所包含，因而成为冗余，可以取消（或者叫作：被吸收）。表 1.3 列出吸收规则的几种主要形式。

表 1.3 几种吸收规则及其对偶形式

i	$A+AB=A$	$A\cdot(A+B)=A$
ii	$A+\bar{A}B=A+B$	$A\cdot(\bar{A}+B)=AB$
iii	$AB+\bar{A}C+BC=AB+\bar{A}C$	$(A+B)(\bar{A}+C)(B+C)=(A+B)(\bar{A}+C)$

说明于下：

i. 原变量的吸收：$A+AB=A$

此式指出：逻辑表达式如果可以分解成两部分，一部分含有另一部分作因子，则含因子的项可被吸收。

例如：$AB+CD+AB\bar{C}D(E+F)=AB+CD$

ii. 反变量的吸收：$A+\bar{A}B=A+B$

此式指出：逻辑表达式如果可以分成两部分，一部分含有另一部分的非函数作因子，则此非函数因子可以被吸收。例如：$A+\bar{A}BC+DE=A+BC+DE$；

iii. 混合变量的吸收：$AB+\bar{A}C+BC=AB+\bar{A}C$

此式表明：当变量 A 及 $\bar{A}$ 的乘积因子分别为 B、C 时，则由 B、C 组成的乘积项是冗余的，可以被吸收。例如：$XYZ+(\overline{XY})W+ZW=XYZ+\overline{XY}W$

表 1.3 中所列各公式不难用以下两种方法证明：

i. 直接用已知定理推导出来。例如：

$$A+AB=A(1+B)=A\times 1=A$$
$$A+\bar{A}B=A+AB+\bar{A}B \qquad (\text{根据 } A+AB=A)$$
$$=A+B(A+\bar{A})=A+B$$

ii. 穷举法：列举变量真值的全部可能组合，证明在所有组合下该式的两边都相等。例如，为了证明 $AB+\bar{A}C+BC=AB+\bar{A}C$ 可作表如表 1.4。表的左方列举变量 A、B、C 取值的各种组

此为试读，需要完整PDF请访问：www.ertongbook.com

表 1.4

A	B	C	AB	$\bar{A}C$	BC	$AB+\bar{A}C+BC$	$AB+\overline{\bar{A}C}$
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	1
0	1	0	0	0	0	0	0
0	1	1	0	1	1	1	1
1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	1	0	1	1	1

合；表的中间部分是式中各项的运算结果；表的右方则是上式两边运算的最后结果。可见不论 A、B、C 取值如何组合，上式两边运算结果总一致。这样就证明了该式。

（三）对偶原理 正象电路理论中有对偶关系一样，逻辑电路中也存在对偶关系。对偶原理指出：

一个逻辑表达式如果成立，则把式中的'与'运算和'或'运算互换，所得的新表达式也必定成立。

例如：由分配律 $$A(B+C)=AB+AC$$

根据对偶原理得：

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

后一等式叫作加法形式的分配律。

又例如，由原变量的吸收律 $$A+AB=A$$

根据对偶原理得到它的对偶式： $$A(A+B)=A$$

表 1.3 的右方列出了各种形式吸收规则的对偶式。

如果原式中还含有常量"1"或"0"，则写对偶式时也必须将 1 和 0 互换。例如：

由 $1+A=1$， 按对偶原理得 $0\cdot A=0$

利用对偶原理可以由一个逻辑等式推出另一个逻辑等式，使待证明的逻辑等式减少一半。这个定理可以用数学归纳法加以证明，本书从略。

（四）反演定理（摩根定理）。 这是一个在化简较复杂的电路（特别是含有与非、或非等逻辑门）时很有用的定理。它指出：

两个（或两个以上）变量的与非运算等于这两个变量的非或运算：

$$\overline{A\cdot B}=\bar{A}+\bar{B}$$

根据对偶原理可立即推论出它的对偶形式：

$$\overline{A+B}=\bar{A}\cdot\bar{B}$$

用文字叙述，即：两个（或两个以上）变量的或非运算等于这两个变量的非与运算。

对两变量情况，定理很容易用穷举法证明；再用数学归纳法或变量置换就能推广到多变量情况。这个定理还可以换一种陈述方式，如下：

设 $F=A\cdot B$ 则 $\bar{F}=\bar{A}+\bar{B}$

设 $F=A+B$ 则 $\bar{F}=\bar{A}\cdot\bar{B}$

它指明构成反函数的方法是：把表达式中变量换成反变量，同时把'与'和'或'两种运算互换。因为定理的这一陈述形式表明对逻辑表达式求反函数的途径，所以叫作反演定理。

（五）香农定理 由反演定理的后一个表述形式可以推演出一个更一般的求反函数方法，叫香农定理：

设F是任一逻辑表达式，则它的反函数可如下求得：把F式中所有变量都换成相应反变量，同时把原式中所有'与'运算和'或'运算都加以互换。

实际上，这个定理只不过是多次反复运用反演定理的结果。

例 1.1 $F=A+\overline{B+\bar{C}+\overline{D+\bar{E}}}$，求$\bar{F}$

令$X=B+\bar{C}+\overline{D+\bar{E}}$，则上式变成：$F=A+\bar{X}$

根据反演定理：$\bar{F}=\bar{A}\cdot X=\bar{A}(B+\bar{C}+\overline{D+\bar{E}})$

再令 $Y=B+\bar{C}$，$Z=D+\bar{E}$，则上式变成：$\bar{F}=\bar{A}(Y+\bar{Z})$对括弧内部分应用反演定理，得：

$$\bar{F}=\bar{A}(\overline{\bar{Y}\cdot Z})$$

但，由反演定理，$\bar{Y}=\bar{B}\cdot C$　　$Z=D+\bar{E}=\overline{\bar{D}\cdot E}$

所以最后得：$\bar{F}=\bar{A}[\overline{\bar{B}\cdot C(D+\bar{E})}]=\bar{A}[\overline{\bar{B}\cdot C\cdot\overline{\bar{D}\cdot E}}]$

最后结果和原式比较可见，可以直接对原式应用香农定理得出$\bar{F}$式。

例 1.2 已知$F=\bar{A}\bar{B}+CD$，求$\bar{F}$。

根据香农定理，$\bar{F}=(A+B)(\bar{C}+\bar{D})$

例 1.3 证明图 1.4 上两个电路的逻辑功能相同。

对图 1.4(a)　$P=E+F$，$E=AB$，$F=CD$

所以　$P=AB+CD$

对图 1.4(b)　$P=\overline{E'\cdot F'}=\bar{E}'+\bar{F}'$

但　$E'=\overline{AB}$，$F'=\overline{CD}$

所以　$P=\overline{\overline{AB}}+\overline{\overline{CD}}=AB+CD$

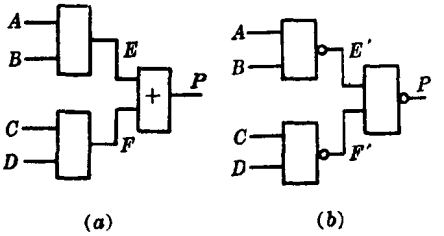


图 1.4

可见两个电路的逻辑功能相同，都能实现与或逻辑。用两级与非门来实现与或逻辑是构作数字电路时常用的方法。

例 1.4 图 1.5(a)是熟知的可控 RS 锁定器。在中规模组件中为了工艺制造便利，常用两个与或非门代替它，如图 1.5(b)所示。试证明这两个电路的逻辑功能等效。

图 1.5(b)中：

$$Q=\overline{\bar{Q}+\overline{CP\cdot R}} \tag{i}$$

$$\bar{Q}=\overline{Q+\overline{CP\cdot S}} \tag{ii}$$

图 1.5(a)中：

$$Q=\overline{\bar{Q}\cdot B}=\overline{\bar{Q}\cdot\overline{CP\cdot S}} \tag{iii}$$

$$\bar{Q}=\overline{Q\cdot A}=\overline{Q\cdot\overline{CP\cdot R}} \tag{iv}$$

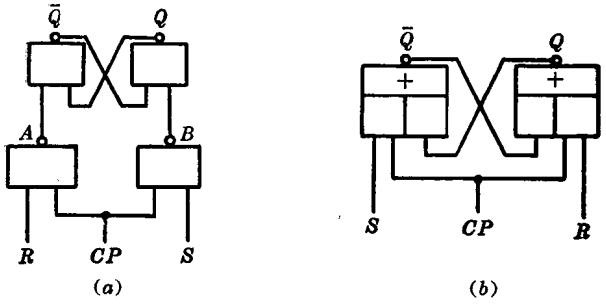


图 1.5

对(iii)式先求反一次，再应用反演定理得：

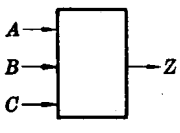
$$\bar{Q}=\bar{Q}\cdot\overline{CP\cdot S}=\overline{Q+CP\cdot S}$$

可见与(ii)式一致。同样可证(iv)式和(i)式一致。

1.4 逻辑函数的表示方法

为了分析研究的便利，同一事物往往有不只一种表示法以便突出它的不同特点。逻辑函数也是这样。表示逻辑函数的常用方法有三种：真值表、图形表示和代数表示式。必须能熟练地在三种表示方法间进行互换。

(一) 真值表 n 个输入变量可组合成 2^n 种不同状态。把对应于每一输入状态下的输出值一一列举出来就构成所谓“真值表”。例如，图 1.6(a)电路有三个输入端，组合起来可以有八种不同状态。把这八种输入状态下的输出列举出来，便得到图 1.6(b)所示真值表。



(a)

A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

(b)

图 1.6

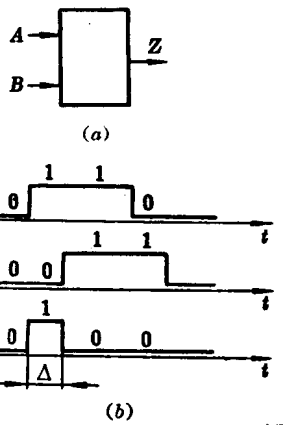
实际工作中真值表要根据任务要求来列写。

例 1.5 一个走廊灯要从三个地点分别独立地进行控制。试列写其真值表。

用 A、B、C 代表三个开关，当它们的取值为 0 时代表开关断开，为 1 时代表开关闭合。再用 $Z=1$ 代表灯亮，则得真值表如表 1.5。

表 1.5

A	B	C	Z	说　明
0	0	0	0	三开关都断开，则灯不亮
1	0	0	1	任一开关闭合，则灯亮
0	1	0	1	
0	0	1	1	
1	1	0	0	任两开关闭合，则灯又不亮
1	0	1	0	
0	1	1	0	
1	1	1	1	三个开关都闭合，则灯又亮



A	B	Z
0	0	0
1	0	1
1	1	0
0	1	0

(c)

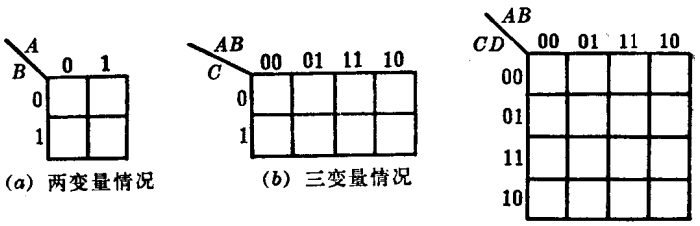
图 1.7

在信号处理任务中，真值表往往可根据对信号波形的要求作出。例如，图 1.7(a)中，输入 A、B 是两个脉宽相等但 B 比 A 延迟 Δ 秒的脉冲。要求输出 Z 反映其延迟时间。图 1.7(b)是根据要求画出的波形图。根据它可得出图 1.7(c)的真值表。

有时输入的 2^n 种组合中有一些未加说明，或实际上不会出现，称为“未定义状态”或“无所谓状态”。例如，表 1.6 是用四个逻辑变量表示十进制数的 8421 编码表。四个逻辑变量本来可以产生十六种组合状态，但其中 $ABCD$=**1010, 1011, 1100, 1101, 1110, 1111** 六个状态实际上未被利用，它们不会出现，便属于无所谓状态。以后将会看到，可以利用这些不会出现的状态来简化设计出来的电路。

表 1.6

十进制	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1



(a) 两变量情况　(b) 三变量情况　(c) 四变量情况

图 1.8

(二) 图象表示　可以用作图的办法较简捷直观地把真值表表示出来。目前应用较广泛的作图法是卡诺图。卡诺图实质上是把对应于输入变量不同组合下的输出值用阵列图表示出来(图 1.8)。阵列的特点是：

1) 阵列图中每个单元代表输入的一种组合，并且把对应的输入组合注明在阵列的上方及左侧。例如，图 1.8(c)左上角单元的对应输入是 $ABCD$=**0000**，右上角单元的对应输入则是 $ABCD$=**1000**。这些输入组合可以通俗地比喻作各单元的“房号”。

2) 编制“房号”的原则是：相邻的两行或两列，其房号只容许有一位互异。仍以图 1.8(c)四变量情况为例。第二列相应于编号 AB=**01**，而相邻的第一列编号是 AB=**00**，二者相比较：A 相同，只有 B 不同；第三列是 AB=**11**，与第二列相比较：B 相同而 A 不同，都是只有一位互异。在横向(行方向)上也有类似特点。注意：阵列的最左端和最右端的两列之间，上下端的两行之间也应看作相邻，它们的“房号”也只有一位互异。

当输入变量多于四个时要想保持上述编号原则就有些困难。此时需要把几个四变量阵列结合起来构成卡诺图。图 1.9(a)是五个输入变量 $ABCDE$ 时的卡诺图，两个四变量阵列分别属于 E=**0** 和 E=**1** 的情况。图 1.9(b)是六个输入变量 $ABCDEF$ 时的卡诺图，四个四变量阵列分别属于 EF=**00, 01, 10, 11** 时的情况。

3) 把真值表中各输出值注明在相应的单元中。如果输入属于无所谓状态则记以 $\varnothing$ 号，$\times$ 号或 $-$ 号。图 1-10(a) 是根据真值表表 1.5 作出的卡诺图。图 1-10(b)则标明了表 1.6 中的

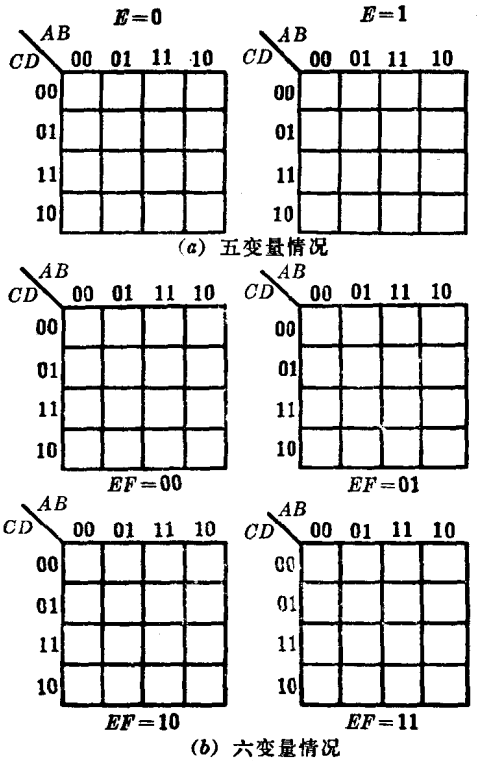


图 1.9

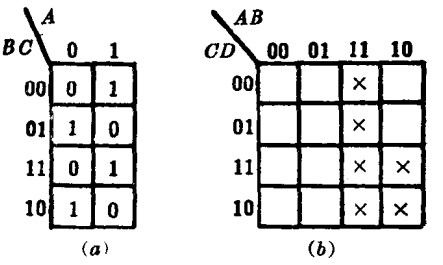


图 1.10

六个无所谓状态。

（三）逻辑代数表达式 真值表所表示的逻辑函数也可以用逻辑代数式来表示。表示式通常有四种标准形式:

1）输出原函数的与或表示式。将真值表中输出 $=1$ 的各状态表示成全部输入变量（包括正变量和反变量）的与函数。（例如表 1.5 中，当 $ABC=100$ 时 $Z=1$，可写成 $Z=A\bar{B}\bar{C}$。因为此式只当 $A=\bar{B}=\bar{C}=1$ 时，即 $ABC=100$ 时 Z 才等于 1。）并把总输出表示成这些与项的或函数。例如，对表 1.5 有:

$$Z=A\bar{B}\bar{C}+\bar{A}B\bar{C}+\bar{A}\bar{B}C+ABC \tag{1-6}$$

这种表示法称为“与或”或“积之和”表示式。它表明当四种输入组合中有任意一个（或一个以上）成立，输出便等于 1。表示式中每个与项都包括全部输入变量在内，称为最小项。它的图形含义就是卡诺图的一个基本单元。

2）输出反函数的与或表示式。将真值表中输出 $=0$ 的各状态表示成全部输入变量的与函数。（例如表 1.5 中当输入 $ABC=110$ 时 $Z=0$，可写成 $\bar{Z}=AB\bar{C}$。因为此式只当 $A=B=\bar{C}=1$，即 $AB\bar{C}=110$ 时才有 $\bar{Z}=1$，也就是 $Z=0$。）并把总输出也表示成这些与项的或函数。例如，对表 1.5 有:

$$\bar{Z}=AB\bar{C}+A\bar{B}C+\bar{A}BC+\bar{A}\bar{B}\bar{C} \tag{1-7}$$

要注意的是，这时输出应写作反变量 $\bar{Z}$，因为当四种输入组合中有任意一个成立时输出将等于 0，而不是等于 1。

3）输出原函数的或与表示式 对(1-7)式应用香农定理，得:

$$Z=(\bar{A}+\bar{B}+C)(\bar{A}+B+\bar{C})(A+\bar{B}+\bar{C})(A+B+C) \tag{1-8}$$

这时输出被表示成若干因子的乘积，其中每一个因子是用全部输入变量（包括正变量和反变量）的或函数表示的，称为“或与”表示式或“和之积”表示式。其中每一个或因子都包括着全部输入变量，称为最大项。举例说明它的代数含义如下：表 1.5 中有 $ABC=110$ 时 $Z=0$，即 $\bar{Z}=AB\bar{C}$。

改成最大项表示法是：$Z=\bar{A}+\bar{B}+C$。它的含义和 $\bar{Z}=AB\bar{C}$ 完全相同。因为它是或函数，只当全部或项都是 **0**，即 $\bar{A}=\bar{B}=C=0$（也就是 $AB\bar{C}=1$）时输出 Z 才等于 **0**。

4）输出反函数的或与表示式。同理，对(1-6)式应用香农定理可得：

$$\bar{Z}=(\bar{A}+B+C)(A+\bar{B}+C)(A+B+\bar{C})(\bar{A}+\bar{B}+\bar{C}) \tag{1-9}$$

上述四种表示式完全等效。实际工作中究竟采用哪一种，一方面决定于哪一种表示式较简单，另一方面也决定于组件生产的实际情况。例如，生产中的 TTL 组件多为正逻辑与非门，因此采用(1-6)式或(1-7)式较方便。与或逻辑可用两级与非门的串级来实现，如图 1.11(a) 可实现(1-6)式。如果采用和之积表示式就要用两级或非门来实现，如图 1.11(b)可实现(1-8)式。

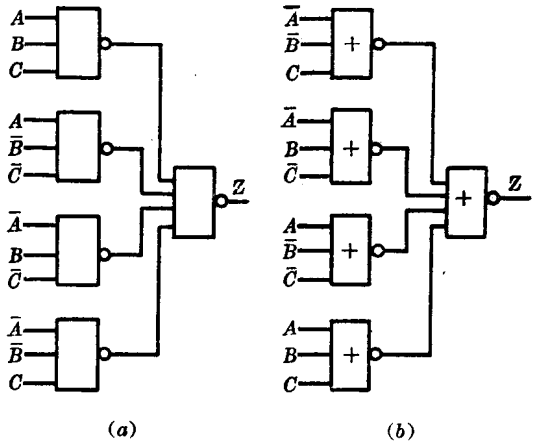


图 1.11

最小项与或表示式和最大项或与表示式是逻辑函数的两种最基本表示法。为了书写便利，往往按下述规则缩写：

最大项用 M 表示，最小项用 m 表示。

变量组合用二进制数的对应十进制数表示。例如：$X_1\bar{X}_2\bar{X}_3X_4$ 是二进制的 **1001**，也就是十进制的 9，因此写作 m_9。而 $X_1+\bar{X}_2+\bar{X}_3+X_4$ 写作 M_9。

最小项之和用 Σ 表示。最大项之积用 Π 表示。

按照上述规则，(1-6)式可缩写成

$$Z=\Sigma(m_1, m_2, m_4, m_7) \text{ 或 } \Sigma m(1, 2, 4, 7)$$

而(1-8)式可缩写成：

$$Z=\Pi(M_1, M_2, M_4, M_7) \text{ 或 } \Pi M(1, 2, 4, 7)$$

例 1.6 将下列各四变量逻辑函数用卡诺图表示。

(a) $f(W, X, Y, Z)=\Sigma m(1, 6, 10, 13, 15)$

(b) $g(A, B, C, D)=\Pi M(2, 6, 8, 10)$

将(a)式各最小项用逻辑式表示出来，如图 1.12(a)，就不难作出其卡诺图如图 1.12(b)。

m	二进制表示	逻辑表示式
1	0001	$\bar{W}\bar{X}\bar{Y}Z$
6	0110	$\bar{W}XY\bar{Z}$
10	1010	$W\bar{X}Y\bar{Z}$
13	1101	$WX\bar{Y}Z$
15	1111	$WXYZ$

(a)

YZ \ WX	00	01	11	10
00	0	0	0	0
01	1	0	1	0
11	0	0	1	0
10	0	1	0	1

(b)

图 1.12

此为试读，需要完整PDF请访问：www.ertongbook.com