

TTL

集成电路

电气特性及应用

上海人民出版社

TTL 集 成 电 路

(电气特性及应用)

巴 益 久 译

上 海 人 民 出 版 社

DESIGNING WITH TTL INTEGRATED CIRCUITS

Robert L. Morris & John R. Miller

McGraw-Hill Book Company

New York 1971

TTL 集 成 电 路

(电气特性及应用)

巴 益 久 译

上海人民出版社出版

(上海绍兴路5号)

新华书店上海发行所发行 上海市印刷六厂印刷

开本 850×1156 1/32 印张 11 字数 289,000

1975年1月第1版 1975年1月第1次印刷

印数 1—17,000

统一书号: 15171·149 定价: 0.68 元

内 容 提 要

本书专门讨论 TTL (晶体管-晶体管逻辑) 集成电路, 由[美]德克萨斯仪器公司集成电路应用组编写。全书十二章。主要介绍 SN 54/74 系列小规模、中规模 TTL 集成电路的基本原理, 电气特性, 电路类型, 以及结合该公司的基本产品介绍在电子计算机、自动控制、仪表等数字系统中的应用, 并列举了许多实用线路, 对使用中所遇到的问题和解决方法作了较具体的分析。此外, 还以较多篇幅介绍数字逻辑的基本知识和逻辑设计技术。

本书可供从事电子计算机、自动控制、仪表等方面工作的工人、科技人员以及有关院校师生参考。

DS87/53-14

译 者 序

在批林批孔运动推动下,我国电子工业发展迅速。国产第三代电子计算机已在生产、科研部门运用。电子计算机、自动控制、数字仪表等都大量使用 TTL 集成电路。目前, TTL 集成电路是我国数字集成电路的主要产品之一。

本书是美国德克萨斯仪器公司集成电路应用组编写的。主要内容是介绍 SN 54/74 系列小规模、中规模 TTL 集成电路的工作原理,电气特性,电路类型及其应用等。书中所讨论的 TTL 电路种类较多,从基本的门电路到各种组合时序线路,包括触发器、译码器、寄存器和运算单元等。此外,本书还以较多的篇幅通俗地介绍数字逻辑的基本知识和逻辑设计技术,在介绍这些内容时,是以该公司的基本产品为背景,列举了不少的实用线路,并对应用 TTL 集成电路所遇到的具体问题和解决方法作了较具体的分析。本书编写的明显目的,是推广其产品。我们感到该书的内容较实用,在技术上可以不受该公司产品的限制,对应用国产的 TTL 集成电路进行设计有一定参考价值。因此,在有关单位的支持下,我们翻译了该书,并根据译者的理解对文字、图表及公式中的错误作了一些订正,供从事集成电路设计、生产和应用等方面工作的同志和相应专业的师生参考,批判地吸取其中有益的东西,为我所用。由于集成电路是项新技术,发展较快,读者在参阅本书时应注意有关方面的新发展。

由于我们的水平所限,译文中一定存在不少缺点和错误,希读者批评指正,并对本书翻译过程中曾给予帮助的同志表示感谢。

巴益久 1974 年 9 月

目 录

译者序

第一章	数字逻辑的基本知识	1
1.1	二进制系统	1
1.2	逻辑电路	3
1.3	逻辑门的应用	6
1.4	逻辑定义	9
1.5	各种逻辑电路的比较	10
1.6	集成电路的集成度	15
第二章	54/74 系列概况	17
2.1	典型特性	17
2.2	标准 TTL 54/74 系列	17
2.3	低功耗 TTL 54L/74 L 系列	21
2.4	高速 TTL 54H/74H 系列	22
2.5	肖特基箝位 TTL 门 54S/74S 系列	24
第三章	54/74 系列的电路分析和特性	26
3.1	基本的 TTL 工作原理	26
3.2	TTL 的特点	27
3.3	电路参数	28
3.4	特殊门的电路特性	49
3.5	TTL 负载规则(附录)	63
第四章	工作范围的扩大	79
4.1	集成电路元件	79
4.2	器件对外部因素的响应	81
4.3	输入端和输出端的电压击穿	91

第五章 噪声的分析	95
5.1 噪声的类型及其抑制方法	95
5.2 屏蔽	96
5.3 接地和去耦	97
5.4 串扰	106
5.5 传输线的反射	110
第六章 组合逻辑设计	121
6.1 基本函数	121
6.2 公设和定理	122
6.3 逻辑表达式	123
6.4 简化和最小化	125
6.5 实用逻辑门	128
6.6 逻辑电路的分析	129
6.7 逻辑表达式的实现	133
6.8 基本逻辑函数的实现	135
6.9 组合逻辑应用	149
第七章 触发器	176
7.1 触发器的类型	178
7.2 54/74 系列触发器	189
7.3 触发器的应用	192
第八章 译码器	201
8.1 译码器原理	201
8.2 54/74 系列的译码器和译码器/驱动器	212
8.3 译码器的应用	227
第九章 运算单元	232
9.1 二进制加法	232
9.2 并行二进制加法器	234
9.3 串行二进制加法器	235
9.4 TTL 54/74 系列运算单元	235

9.5	计算机运算采用的二进制表示法	240
9.6	用二进制表示十进制加法和减法	246
9.7	快速二进制加法	256
9.8	加法器在二进制数码转换中的应用	262
第十章	计数器	265
10.1	行波计数器	265
10.2	同步计数器	271
10.3	54/74 系列计数器	279
10.4	计数器的构成及其应用	297
第十一章	移位寄存器	309
11.1	SN 54/74 系列移位寄存器	310
11.2	移位寄存器型计数器与发生器	316
11.3	移位寄存器的其他应用	331
第十二章	其他应用	333
12.1	一种简单的二进制乘法器	333
12.2	12 小时的数字时钟	337
12.3	串行格雷码-二进制数的转换	339
12.4	模 360 加法器	341

第一章 数字逻辑的基本知识

最初的电子计算机,采用的是十进制系统,这就要求每位有十种不同的电平,因而很烦琐。规定和维持十种电平很困难,所以采用仅有两种电平或数字(0和1)的简单的二进制系统来代替十进制系统。在二进制运算中,一个量值大小的判定方式,采用晶体管线路,由其输出电压的高低来实现是相当容易的。同时,晶体管在小于1微秒的时间内可以从一种状态转换到另一种状态,因此,它每秒至少可以作出一百万次逻辑判定。

数字计算机逻辑网络执行的基本操作是加法运算,减法、乘法和除法都是变换成加法过程来实现的。例如: 15×5 ,数字计算机是把数15加五次。虽然这种操作要求许多二进制线路,但是系统的简单性以及晶体管和集成电路的经济性,使二进制数字计算机得到迅速发展。

数据以二种不同电平的脉冲送入数字计算机,而信息以脉冲的数量和时间间隔来表示。在大多数计算机中,执行的是二进制运算,它是把以十进制形式输入计算机的数据转换成等值的二进制形式,再将结果转换成十进制形式输出。

1.1 二进制系统

要理解数字集成电路的工作,就要求熟悉二进制系统及其在逻辑判定式中的应用。在二进制语言中,第一种状态(断开)称为“0”;第二种状态(接通)称为“1”。

在十进制系统中,在个位可以计数到9,若再一次计数时,个位数便回到0,并且向第二位进1,这样表示个位数已全部计过一次,得到10。至于二进制,它的计数方法基本上与十进制一样,只不过仅用到数0和1,计1之后,个位数就用过了,再下一次计数

时就必须进到第二位,以表示第一位的全部计数单元已计过一次。因而在十进制系统中的数 2, 在二进制中被表示为 10 (称为 1-0, 不是十)。下一次计数即由 0 变为 1, 得到 11(1-1), 相当于十进制中的 3。这样, 又一次用过了全部计数单元, 再下一次计数, 两位都必须回到 0, 而在第三位上置 1, 得到二进制的 100, 相当于十进制的 4。

表 1.1 十进制转换成二进制

十 进 制	二 进 制	十 进 制	二 进 制
0	0	10	1010
1	1	11	1011
2	10	12	1100
3	11	13	1101
4	100	14	1110
5	101	15	1111
6	110	16	10000
7	111	32	100000
8	1000	64	1000000
9	1001	128	10000000

表 1.1 列出与某些十进制数等值的二进制数。由此可看出: 二进制 10 相当于十进制 2, 即 2^1 ; 二进制 100 相当于十进制 4, 即 2^2 ; 二进制 1000 相当于十进制 8, 即 2^3 ; 二进制 10000 相当于十进制 16, 即 2^4 。由此可见, 二进制数每增加一位, 相当于 2 的幂加 1。这条规则在将二进制数转换成等值的十进制数时很有用。例如: 将二进制数 11010 转换成十进制数, 这个二进制数等于 $2^4 + 2^3 + 0 + 2^1 + 0$, 或 $16 + 8 + 0 + 2 + 0$, 即等于十进制数 26。反之, 十进制数可以利用反复地减去可能最高的 2 的幂的办法转换成二进制数。例如: 十进制数 26, 第一次减 16 (2^4), 在二进制中即是 10000, 从余下的 10 中减 8 (2^3), 即二进制的 1000, 余下的 2 即是二进制的 10, 这些二进制数相加 $10000 + 1000 + 10$, 得 11010。

显然, 二进制数比与它等值的十进制数需要更长的数字序列,

特别对于较大的数值更是这样。但是,由于电子数字计算机每秒能处理几百万次简单的加法,所以二进制数的字长不会引起严重的问题。

1.2 逻辑电路

各种逻辑判定是通过逻辑电路来实现,以便获得一组给定条件的问题的逻辑解答。为了实现逻辑判定操作,一般要采用下列三种基本逻辑电路(称为门):“或”电路,“与”电路和“非”电路。

“或”电路 这种电路的基本形式有两个或两个以上的输入端和一个输出端。输入和输出可以各自为“0”或“1”状态。该电路设计成这样:当任意一个输入状态为“1”时,输出便为“1”。即在输入端 A 、 B 或 C 中任意一个输入端为“1”时,输出是“1”。电路可用图 1.1(a)的模拟图来说明。电源通过三个并联的开关加到一只灯 L 上,这些开关作为灯的三个输入,灯的状态表示电路输出。

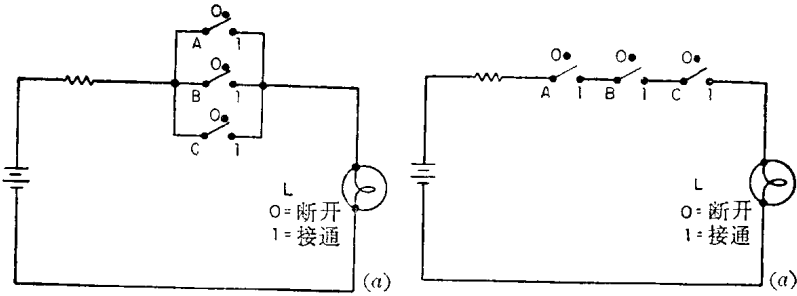
如果我们规定:开关断开为输入“0”状态,灯不亮为输出“0”状态;开关接通为输入“1”状态,灯亮为输出“1”状态。那么我们可以列出各种不同的开关组合状态以及由此产生的输出状态,这种表格叫做真值表,示于图 1.1(b)。从真值表中可以看到:要灯不亮(输出“0”状态),所有开关都必须断开(输入“0”状态)。

这种形式的电路称为“或”门,用图 1.1(c)所示的符号来代表。该图表示了一个有三个输入端的“或”门。因此,“或”门能用来构成这样的逻辑判定,几个输入中是否至少有一个在“1”状态。

“与”电路 这种电路的基本形式也有几个输入端和一个输出端,但是只有所有的输入同时都为逻辑“1”状态时,输出才是逻辑“1”状态。这种电路原理可用图 1.2(a)来描述,只有全部开关 A 、 B 和 C 都同时接通时,灯 L 才亮;倘若其中任一开关断开,灯就不亮。采用与前面相同的表示法,“与”电路的真值表示于图 1.2(b),符号如图 1.2(c)所示。因此,“与”门可用来构成这样的逻辑判定,即几个输入是否同时都为“1”状态。

这里,我们简单地提一下扇入和扇出问题。门的输入端数叫

做扇入,在上述的例子中,每个门的扇入为3。单独一个门的输出能够驱动下一级门的数量称为扇出。一个门只有一个输出信号,但是可以要求这个输出信号供给其他几个逻辑门。



A	B	C	L
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

A	B	C	L
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

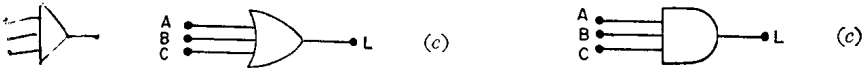


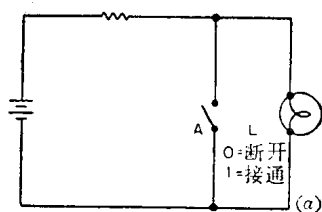
图 1.1 “或”电路

图 1.2 “与”电路

(a) 模拟图; (b) 真值表; (c) 符号

“非”电路 这种电路的基本形式只有一个输入端和一个输出端,并且其输出状态总是和输入状态相反。研究一下图1.3(a),当开关断开(“0”)时,电流流过灯,灯亮(“1”)。倘若开关接通,那么电流流过开关而不经灯,灯熄灭。使输出状态和输入状态相反的这种操作称为反相,为了实现这种操作而设计的电路称为反相器。简单的真值表示于图1.3(b),它的符号示于图1.3(c)。

“或非”和“与非”电路 一个“非”电路可以和一个“或”门或“与”门组合起来,使反相和门的功能同时存在。“非”电路和“或”门的组合称为“或非”门,可用图1.4(a)的灯电路模拟图来说明,倘



A	L
0	1
1	0

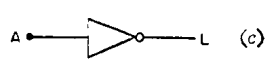
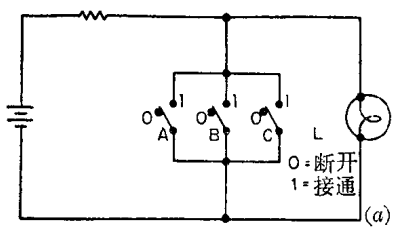


图 1.3 “非”电路

(a) 模拟图; (b) 真值表; (c) 符号

若其中任一开关是在“1”状态,那么灯是在“0”状态,其真值表示于图 1.4(b),符号示于图 1.4(c)。

同样,一个“非”电路和“与”门的组合称为“与非”门,见图



A	B	C	L
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

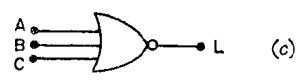
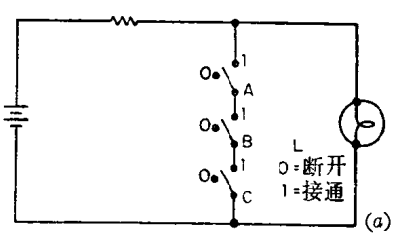


图 1.4 “或非”电路



A	B	C	L
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

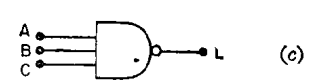


图 1.5 “与非”电路

(a) 模拟图; (b) 真值表; (c) 符号

1.5(a)。当所有开关是在“1”状态时,灯是在“0”状态。“与非”电路的真值表示于图 1.5(b),符号示于图 1.5(c)。

1.3 逻辑门的应用

为了说明逻辑门的应用,首先研究二个二进制数 A 和 B 相加的操作。考虑一种最简单的情况, A 和 B 都是由一位二进制数组成,或者为“0”,或者为“1”。加法线路的逻辑图示于图 1.6(a),整个真值表示于图 1.6(b)。该线路的二个输入 A 和 B 分别接“与”门, A 接“与₁”, B 接“与₂”; A 和 B 还通过反相器 I_1 和 I_2 送到相应的门去。因此,当 A 是“1”时,输入到“与₁”门是“1”,而输入到“与₂”门是“0”;当 A 是“0”时,输入到“与₁”门是“0”,到“与₂”门是“1”。二个“与”门的输出接到一个“或”门,由“或”门的输出得

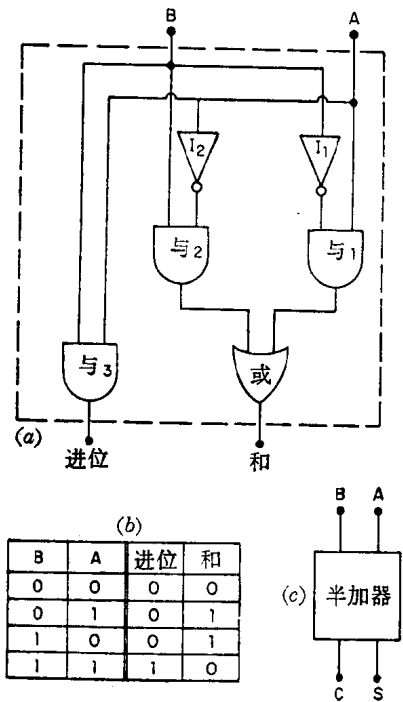


图 1.6 半加器

(a) 逻辑图; (b) 真值表; (c) 符号

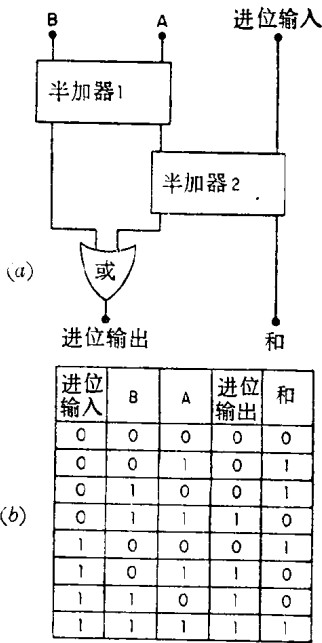


图 1.7 全加器

(a) 逻辑图; (b) 真值表

到和“ S ”。 A 和 B 还直接输入到第三个“与”门(“与₃”), 其输出端给出进位 C 。

对此线路的操作, 有四种可能情况:

1. $A=0; B=0$ 。每个“与”门的输入至少有一个为“0”, 故没有一个“与”门能输出“1”。因此, 和及进位都为“0”, 故得出答案 00。

2. $A=1; B=0$ 。 A 端输出“1”送到“与₁”门, B 端的“0”经 I_1 反相输出另一个“1”送到“与₁”门, 这样, “与₁”门的两个输入都是“1”, 因此, 输出一个“1”到“或”门; 输入到“与₂”门的都是“0”, 所以该门输出为“0”。因为“或”门的输入中有一个为“1”, 故由其输出的和是“1”。由于进位门“与₃”的两个输入中有一个是“0”, 因而进位输出是“0”。因此, 得出答案 01。

3. $A=0; B=1$ 。其操作和上述情况 2 相同, 但输入到“与₁”和“与₂”的状态相反。答案还是 01。

4. $A=1; B=1$ 。“与₁”、“与₂”门都不能给出“1”输出, 因为它们从反相器得到的一个输入是“0”, 所以和是“0”。但是进位门“与₃”的输入都是“1”, 故给出一个进位“1”输出。因此, 答案是 10。

这种线路可视为具有两个输入和两个输出的基本逻辑部件, 如图 1.6(a) 虚线框所示, 其符号如图 1.6(c) 所示, 通常称之为半加器, 这是因为它只能作一位数相加。如果要进行两个多位数相加时, 就必须使这种线路能够接收并加上前一位来的进位。为了做到这一点, 必须有一个全加器线路。构成全加器的一种方法是利用二个半加器, 如图 1.7(a) 所示。第一步 A 和 B 相加, 第二步将形成的和与前一位来的进位相加, 得到最终的和。将两个半加器的进位输出送到一个“或”门上, 它的输出给出最终的进位。全加器的真值表示于图 1.7(b)。稍加分析就可看出, 两个半加器的输出端不可能同时存在进位输出。

多位二进制数相加可用一串并行的全加器。第一位没有进位输入, 可以是一个半加器。两个 3 位二进制数 $A_3A_2A_1$ 和 $B_3B_2B_1$

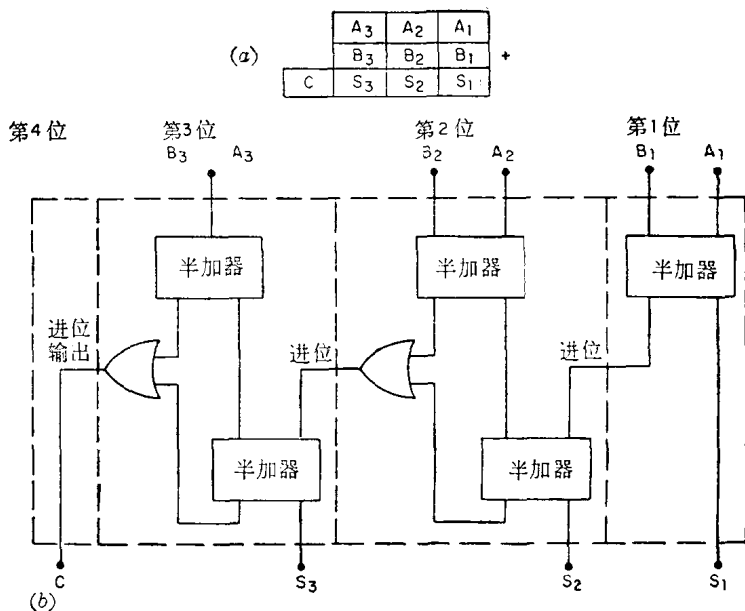


图 1.8 三位并行加法器

(a) 加法表; (b) 逻辑图

相加的加法表如图 1.8(a) 所示, 系统框图如图 1.8(b) 所示。

统计一下这种加法部件所需门的数量是有意义的。如图 1.7 所示的一个全加器, 同一位两个二进制数相加就需要六个“与”门、三个“或”门和四个反相器, 总共十三个门。现代计算机必须处理十位左右的十进制数, 即 10,000,000,000 或 2^{33} , 相当于 34 位二进制数。两个 34 位的二进制数相加, 需要 33 个全加器和一个半加器, 总共需要 435 个门。

只要了解一下用反复相加实现乘法所要求的设备数量和实现其他操作所需的设备数量, 就不难看出为什么在现代数字计算机的运算部件中, 门的数量常常会超过 10000 个。在这种部件里, 将反复使用相同类型的门: 所有的“与”门可以是相同的, 所有的“或”门、反相器也同样如此。采用集成电路工艺, 可以把许多相同的线路制作在一块硅晶片上作为一种逻辑电路的器件。每块硅片具有相同的线路性能, 这样使用方便, 而造价降低。

在讨论二进制加法时,我们引入了代表“与”门和“或”门的框框,而总的功能将取决于它们的互连方式,这就是逻辑电路的重要设计课题。只要这些门能满意地表示二进制数“0”和“1”两种工作状态,那么,逻辑系统设计就可以在图纸上完成。从逻辑系统观点来看,不管框框代表的是什么——继电器、电子管、磁芯、晶体管或集成电路,所有的逻辑功能将是相同的。而究竟采用哪一种器件则取决于下列一些因素:如价格、体积、功耗、速度和可靠性。

1.4 逻辑定义

在大多数现代逻辑系统中,用电平来表示逻辑“1”和“0”。在数字系统中,关于这些逻辑电平的定义,公认的规则是:正逻辑定义高电平为逻辑“1”状态,低电平为逻辑“0”状态;负逻辑正好相反,高电平表示“0”,而低电平表示“1”。

从一种逻辑定义变到另一种定义,实际上是对全部逻辑函数取反。例如:“与”变成“或”,“或非”变成“与非”……等等。变换逻辑定义(正或负)的最简单的方法是在器件的真值表中将所有的“0”用“1”代替,而所有的“1”用“0”代替,然后确定所形成的逻辑函数(第六章“组合逻辑设计”对这个问题作了一些详细说明)。

选择正逻辑或负逻辑,取决于逻辑设计者的习惯,任一种并不比另一种有什么优越的地方,而大多数逻辑设计者和关于逻辑设计的教科书,都采用正逻辑。本书全部采用正逻辑。然而,为了使特定逻辑元件的性能数据尽可能通用,倾向于在真值表中指明 H (高电平)或 L (低电平),而不是“1”和“0”。这种表格称为功能表。这比称为真值表的更为恰当,但是仅由于措辞习惯,流行后一种名称。

功能表中采用 H 和 L ,就无需特别说明真值表是采用正逻辑还是负逻辑,但是它会使通常使用严格的布尔术语(0、1)表示法的人们一时感到不习惯。在纯逻辑理论中并没有“正”逻辑和“负”逻辑这种术语,只有当考虑逻辑功能的具体实现时,才有必要考虑逻辑定义。